Robust and Generalizable Knowledge Acquisition from Text

by

Wenxuan Zhou

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

August 2023

To my family and friends,
who support me during my Ph.D. journey.

Acknowledgements

I express my heartfelt gratitude to my Ph.D. advisor, Prof. Muhao Chen, for his guidance, support, and mentorship throughout my Ph.D. journey. His expertise in NLP and knowledge bases have been invaluable in shaping my research agenda and in guiding my technical development. In addition, I would like to extend my sincere appreciation to Prof. Yangqiu Song and Prof. Xiang Ren for their supervision, valuable insights, and encouragement during my early research years. Their mentorship and guidance were invaluable in helping me establish a strong foundation in the field.

I would like to thank my defense committee members, Prof. Laurent Itti, Prof. Jonathan May, Prof. Tianshu Sun, and Prof. Robin Jia, for their critical feedback and constructive suggestions. I would also like to thank Prof. Fred Morstatter for serving as a member of my qualification committee.

I would like to extend my appreciation to all of my friends at the USC LUKA lab and USC INK lab, including Bill Yuchen Lin, James Huang, Tenghao Huang, Woojeong Jin, Bangzheng Li, Keming Lu, Ehsan Qasemi, Qin Liu, Fei Wang, and Nan Xu, for their support, friendship, and fruitful collaborations. I am also grateful for the contributions of my collaborators from other labs and universities, including Tianqing Fang, I-Hung Hsu, Fangyu Liu, Mingyu Derek Ma, Yiwei Wang, Hongming Zhang, and Huan Zhang.

I am indebted to my mentors during internships, Dr. Heba Elfardy, Dr. Jing Huang, Dr. Hang Li, Prof. Tengyu Ma, Dr. Tristan Naumann, Dr. Qiang Ning, Dr. Hoifung Poon, Dr. Kevin Small, and Dr. Sheng Zhang, for their valuable guidance, support, and the opportunity to work on exciting research problems.

Finally, I want to express my deep appreciation to my parents for their unwavering support, encouragement, and understanding throughout my academic journey.

Table of Contents

Dedication ii

Acknowledgements iii

List of Tables vii

List of Figures ix

Abstract xi

Chapter 1: Introduction 1
- 1.1 Motivations 1
- 1.2 Challenges 2
- 1.3 Thesis Contributions 3
- 1.4 Thesis Overview 4

Chapter 2: Document-level Relation Extraction 7
- 2.1 Introduction 7
- 2.2 Problem Formulation 9
- 2.3 Enhanced BERT Baseline 9
  - 2.3.1 Encoder 10
  - 2.3.2 Binary Classifier 10
- 2.4 Adaptive Thresholding 11
- 2.5 Localized Context Pooling 12
- 2.6 Experiments 14
  - 2.6.1 Datasets 14
  - 2.6.2 Experiment Settings 14
  - 2.6.3 Main Results 16
  - 2.6.4 Results on Biomedical Datasets 16
  - 2.6.5 Ablation Study 17
  - 2.6.6 Analysis of Thresholding 18
  - 2.6.7 Analysis of Context Pooling 18
- 2.7 Related Work 19
- 2.8 Conclusion 20

Chapter 3: Noisy Label Learning 21
- 3.1 Introduction 21
- 3.2 Method 23

3.2.1 Learning Process . . . 23
3.2.2 Co-regularization Objective . . . 24
3.2.3 Joint Training . . . 25
3.3 Tasks . . . 26
3.4 Experiment . . . 26
3.4.1 Datasets . . . 27
3.4.2 Base Models . . . 27
3.4.3 Model Configurations . . . 28
3.4.4 Main Results . . . 28
3.4.5 Noise Filtering Analysis . . . 29
3.4.6 Ablation Study . . . 30
3.5 Related Work . . . 32
3.6 Conclusion . . . 33

Chapter 4: Out-of-distribution Detection . . . 34
4.1 Introduction . . . 34
4.2 Related Work . . . 35
4.3 Method . . . 36
4.3.1 Problem Definition . . . 36
4.3.2 Framework Overview . . . 37
4.3.3 Contrastive Representation Learning . . . 37
4.3.4 OOD Scoring Functions . . . 40
4.4 Experiments . . . 41
4.4.1 Datasets . . . 41
4.4.2 Experimental Settings . . . 43
4.4.3 Main Results . . . 43
4.4.4 Novel Class Detection . . . 44
4.4.5 Analysis . . . 44
4.5 Conclusion . . . 46

Chapter 5: Resolving Knowledge Conflicts . . . 47
5.1 Introduction . . . 47
5.2 Related Work . . . 49
5.3 Method . . . 50
5.3.1 Opinion-based Prompting . . . 50
5.3.2 Counterfactual Demonstration . . . 51
5.4 Experiments . . . 52
5.4.1 Experimental Setup . . . 52
5.4.2 Knowledge Conflict . . . 53
5.4.3 Prediction with Abstention . . . 54
5.4.4 Additional Analysis . . . 55
5.4.5 Case Study . . . 58
5.5 Conclusion . . . 59

Chapter 6: Dataset Annotation by Neural Rule Grounding . . . 60

6.1 Introduction . . . 60
6.2 Problem Formulation . . . 63
6.3 Neural Rule Grounding (NERO) . . . 65
6.3.1 Framework Overview . . . 65
6.3.2 Labeling Rule Generation . . . 66
6.3.3 Relation Classifier (RC) . . . 66
6.3.4 Soft Rule Matcher (SRM) . . . 67
6.3.5 Joint Module Learning . . . 68
6.4 Model Learning and Inference . . . 70
6.4.1 Parameter Learning of NERO . . . 70
6.4.2 Model Inference . . . 71
6.5 Experiments . . . 72
6.5.1 Data Preparation . . . 72
6.5.2 Compared Methods . . . 72
6.5.3 Experiment Settings . . . 73
6.5.4 Performance Comparison . . . 74
6.5.5 Performance Analysis . . . 76
6.5.6 Model Ablation Study . . . 76
6.5.7 Case Study . . . 78
6.6 Related Work . . . 79
6.7 Conclusion . . . 81

Chapter 7: Continual Contrastive Finetuning . . . 82
7.1 Introduction . . . 82
7.2 Related Work . . . 83
7.3 Method . . . 84
7.3.1 Model Architecture . . . 85
7.3.2 Pretraining . . . 85
7.3.3 Finetuning . . . 86
7.3.4 Inference . . . 88
7.4 Experiments . . . 88
7.4.1 Datasets . . . 89
7.4.2 Experimental Setup . . . 89
7.4.3 Main Results . . . 90
7.4.4 Ablation Study . . . 91
7.4.5 Visualization . . . 92
7.5 Conclusion . . . 93

Chapter 8: Conclusion and Future Work . . . 94
8.1 Summary . . . 94
8.2 Future Directions . . . 95

Bibliography . . . 97

List of Tables

2.1 Statistics of the datasets in experiments. 14
2.2 Hyper-parameters in training. 14
2.3 Main results (%) on the development and test set of DocRED. We report the mean and standard deviation of F_1 on the development set by conducting 5 runs of training using different random seeds. We report the official test score of the best checkpoint on the development set. 15
2.4 Test F_1 score (%) on CDR and GDA dataset. Our ATLOP model with the SciBERT encoder outperforms the current SOTA results. 15
2.5 Ablation study of ATLOP on DocRED. We turn off different components of the model one at a time. These ablation results show that both adaptive thresholding and localized context pooling are effective. Logsumexp pooling and group bilinear both bring noticeable gain to the baseline. 17
2.6 Result of different thresholding strategies on DocRED. Our adaptive thresholding consistently outperforms other strategies on the test set. 17
3.1 Data statistics of TACRED and CoNLL03. 27
3.2 F_1 score (%) on the dev and test set of TACRED. ♣ marks results obtained from the originally released implementation. We report the median of F_1 on 5 runs of training using different random seeds. For fair comparison, the CR results are reported based on the predictions from model f_1 in our framework. 28
3.3 F_1 score (%) on the dev and test set of CoNLL03. ♣ marks results obtained using the originally released code. 29
3.4 F_1 score (%) of using different number of models on the relabeled test set of TACRED. 30
3.5 F_1 score (%) of alternative noise filtering strategies on the test set of TACRED. The best results are achieved when $\delta = 2\%$ for both methods. 31
3.6 F_1 score (%) of different functions for $\boldsymbol{q}$ on the relabeled test set of TACRED. . . . 31
3.7 F_1 score (%) under different noise rates on the relabeled set of TACRED. 32
4.1 Statistics of the datasets. 42
4.2 OOD detection performance (in %) of RoBERTa$_{\text{LARGE}}$ trained on the four ID datasets. Due to space limits, for each of the four training ID dataset, we report the macro average of AUROC and FAR95 on all OOD datasets (check Appendix for full results). Results where the contrastive loss improves OOD detection on both evaluation metrics are highlighted in green."w/o $\mathscr{L}_{\text{cont}}$+MSP" thereof is the method in Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1]. 42
4.3 Novel class detection performance. 44
4.4 Accuracy of the trained classifier. 45
4.5 Average OOD detection performance of different distance metrics. 45

4.6 Average OOD detection performance of other pretrained Transformers. 45
5.1 Results (in %) in the knowledge conflict setting. The overall best results are highlighted in **bold**. The best and the second best results in each setting are highlighted in green and orange, respectively. 52
5.2 Results (in %) on RealTime QA. The overall best results are highlighted in **bold**. The best and the second best results in each setting are highlighted in green and orange, respectively. As all prompts achieve perfect accuracy (100%) on the **HasAns** subset, it is not included in the table. 54
5.3 Results (in %) on the filtered evaluation set of natural questions with original (factual) contexts and answers. 57
5.4 Examples of prompts and LLMs' corresponding predictions. In the "Prompt" row, we show and highlight the added parts from different prompting templates including attributed prompts, instruction-based prompts, and opinion-based prompts. 58
6.1 Statistics for TACRED and SemEval datasets. 72
6.2 Performance comparison (in %) of relation extraction on the TACRED and SemEval datasets. We report the mean and standard deviation of the evaluation metrics by conducting 5 runs of training and testing using different random seeds. We use LSTM+ATT ($\mathscr{S}_{\text{matched}} + \mathscr{P}$) as the base model for all semi-supervised baselines and our models. 74
6.3 Performance on predicting unseen relations. NERO applies the learned soft rule matcher on unseen relation rules to make predictions. 77
6.4 Ablation Study of Different Training Objectives on TACRED dataset. We remove each loss term one at a time. 77
6.5 Ablation study of different soft-matching models for NERO on the TACRED dataset. 78
7.1 Probing results (in F_1) on the test set of BioRED and Re-DocRED. 87
7.2 Results on the test set of Re-DocRED. 90
7.3 F_1 on the test set of BioRED. 90
7.4 F_1 on the test set of BioRED with different pretraining objectives. We use MCCL in finetuning. 91

List of Figures

2.1 An example of multi-entity and multi-label problems from the DocRED dataset. Subject entity *John Stanistreet* (in orange) and object entity *Bendigo* (in green) express relations *place of birth* and *place of death*. The related entity mentions are connected by lines. Other entities in the document are highlighted in grey. 8

2.2 An artificial illustration of our proposed adaptive-thresholding loss. A TH class is introduced to separate positive classes and negative classes: positive classes would have higher probabilities than TH, and negative classes would have lower probabilities than TH. 11

2.3 Illustration of localized context pooling. Tokens are weighted averaged to form the localized context $\mathbf{c}^{(s,o)}$ of the entity pair (e_s, e_o). The weights of tokens are derived by multiplying the attention weights of the subject entity e_s and the object entity e_o from the last transformer layer so that only the tokens that are important to both entities (highlighted in light yellow) receive higher weights. 13

2.4 Dev F_1 score of documents with the different number of entities on DocRED. Our localized context pooling achieves better results when the number of entities is larger than 5. The improvement becomes more significant when the number of entities increases. 18

2.5 Context weights of an example from DocRED. We visualize the weight of context tokens $\mathbf{a}^{(s,o)}$ in localized context pooling. The model attends to the most relevant context *born* and *died* for entity pair (*John Stanistreet*, *Bendigo*). 19

3.1 Illustration of our co-regularization framework. The base models are jointly optimized with the task-specific loss from label y and an agreement loss, which regularizes the models to generate similar predictions to the aggregated soft target probability $\boldsymbol{q}$. 22

3.2 F_1 score (%) on the clean set of TACRED. Classifiers trained with our framework are more noise-robust compared to baselines ($\gamma = 0$). 30

4.1 Illustration of our proposed contrastive loss. The contrastive loss seeks to increase the discrepancy of the representations for instances from different training classes, such that OOD instances from unknown classes can be better differentiated. 38

4.2 Visualization of the representations for positive, negative instances in SST2 and OOD ones. The discrepancy between ID and OOD representations is greater on representations obtained with $\mathscr{L}_{\text{margin}}$. 43

5.1 Examples of knowledge conflict and prediction with abstention. LLMs may ignore the provided context and make unfaithful predictions based on their parametric knowledge before Q4 2021. 48

5.2 Memorization ratios across different sizes of InstructGPTs, evaluated in the zero-shot setting using natural questions. 56

5.3 Brier scores across different sizes of InstructGPTs, evaluated in the zero-shot setting of RealTime QA. 57
6.1 Current rule-based methods mostly rely on exact/hard matching to raw corpus and suffer from limited coverage. For example, the rule body of p_3 only matches sentence s_1 but is also similar to s_2 and s_3, which express the same relation as p_3. A "soft rule matching" mechanism is desirable to make better use of the corpus for label generation. 61
6.2 Comparison between previous work and the proposed NERO framework. (A) Bootstrapping. (B) Data Programming. (C) Self-Training. (D) NERO. The neural rule grounding process enables the soft-matching between rules and unmatched sentences using the soft rule matcher. 62
6.3 Overview of the NERO framework. Each unmatched sentence is first annotated by the soft rule matcher (SRM) to generate pseudo labels, and then fed into the relation classifier (RC) to update the model parameters. The whole framework is trained iteratively and jointly, based on multiple loss functions as introduced in Sec. 6.3.5. 65
6.4 Detailed architecture of the soft rule matcher (SRM). The cosine similarity between two embeddings indicates the degree of matching between rules and sentences. . . 68
6.5 The contrastive loss in learning the SRM, which increases the matching score of rules with the same relation type and decrease the matching score otherwise. . . . 69
6.6 Performance w.r.t. different number of rules and human-annotated labels on TACRED. We show different models' F1 scores and number of rules or labels used for training the corresponding model. 75
6.7 Performance of different semi-supervised models trained using various amounts of unlabeled sentences randomly sampled from the raw corpus on TACRED. 77
6.8 Sensitivity analysis of τ, α, β, and γ on TACRED. We report the mean and standard deviation F1 score by conducting 5 runs of experiments using different random seeds. 79
6.9 Output visualization of SRM. Left: attention weights of words and the soft matching scores between a rule and three sentences. Right: cosine similarity matrix between word embeddings learned with the contrastive loss. 80
6.10 Study on label efficiency. Average number of rules / sentences labeled by annotators (dashed line) are shown on the x-axis over the left-hand side; and the performance of models trained with these corresponding labeled rules / sentences (solid line) are shown on the x-axis over the right-hand side. We use NERO and LSTM+ATT as the base model for the labeling rules and sentences, respectively. 80
7.1 F_1 using different temperatures on 1% of BioRED. 91
7.2 F_1 under different percentages of BioRED training data. 92
7.3 Visualization of relation embedding finetuned with different objectives on BioRED. NA instances are shown in grey. 93

Abstract

Knowledge acquisition involves identifying and understanding concepts that are expressed in text, and then inferring and consolidating their relationships. With the ever-increasing volumes of digital text generated every day, the need for automated knowledge acquisition is becoming more crucial. However, real-world knowledge acquisition systems face more complex scenarios where knowledge is often expressed implicitly or within long contexts, and require models that are robust and able to handle these complexities. Additionally, data annotation for knowledge acquisition can be expensive, particularly in high-stakes domains, necessitating the development of efficient methods that minimize human involvement. In this dissertation, we propose solutions to build robust and generalizable knowledge acquisition systems, addressing challenges in: (i) extracting knowledge from long contexts efficiently and effectively; (ii) improving system robustness in various scenarios including noisy annotations, out-of-distribution instances, and knowledge conflicts; and (iii) data-efficient knowledge acquisition, utilizing cheap resources such as labeling rules or unlabeled corpora.

In the first part of this thesis, we study the extraction of knowledge from long contexts such as documents. To handle the long-context problem, we introduce a novel pretrained-attention-guided neural structure that finds context that both entities attend to in an entity pair. We show that our model achieves state-of-the-art performance in both general and biomedical domains.

In the second part of this thesis, our focus is on enhancing the robustness of knowledge acquisition systems. To start with, we investigate the mitigation of noisy labels during training. Since the data used for knowledge acquisition is often sourced from crowdsourcing or distant supervision, it is common for the data to be imperfect. We explore a co-regularization based framework to address this issue effectively. Moreover, many knowledge acquisition systems are typically developed for specialized domains or relations, making them susceptible to erroneous predictions on instances from other domains or relations. To overcome this challenge, we propose an out-of-distribution detection framework to identify such instances during inference and either alert users or redirect them to other dedicated models for improved accuracy. Another challenge is that the world is continually evolving, and the knowledge acquired by these systems may become outdated. We explore techniques for updating the model's predicted knowledge using context retrieval and opinion-based prompts.

Finally, we study how to build knowledge acquisition systems with minimal human effort. Our approach involves weak data creation, where we can generate a large amount of labeled data by utilizing labeling a few rules developed by humans. Then, we investigate enhancing the data efficiency of knowledge acquisition systems by self-supervised learning. To accomplish this, we propose a framework that pretrains a model using a large unlabeled corpus, which can then be fine-tuned with a small number of labeled instances. Additionally, we propose a novel continual

contrastive finetuning framework that aims to minimize the discrepancy between the pretraining and fine-tuning stages.

Chapter 1

Introduction

1.1 Motivations

Knowledge acquisition involves recognizing concepts described in the text, inferring their relationships, and consolidating them. With large amounts of unstructured text generated every day, automated knowledge acquisition is a critical tool for efficiently managing and processing this information. The extracted knowledge plays a crucial role in various NLP tasks such as question answering [2–4], fact verification [5, 6], and dialogue [7, 8]. Moreover, it provides access to expert knowledge for domain-specific research [9–11] and clinical applications [12, 13].

Pretrained language models (PLMs; [14]) have been successful in automatic knowledge acquisition in recent years. Previous works [15–17] mainly focus on knowledge expressed within a single sentence. However, in real-world applications, large amounts of knowledge are expressed through multiple sentences. For example, the knowledge triple *(Bill Gates, founder, Microsoft)* can only be inferred from two sentences in the following Wikipedia example:

> *William Henry Gates III (born October 28, 1955) is an American business magnate, philanthropist, and investor. He is best known for co-founding software giant Microsoft, along with his late childhood friend Paul Allen.*

According to a quantitative study [18], at least 40.7% of knowledge triples in Wikipedia can only be inferred from multiple sentences. Therefore, extending automatic knowledge acquisition beyond individual sentences is a crucial research problem, in order to achieve a higher recall of knowledge contained in text.

Another research problem is the discrepancy between the datasets used in academic research and real-world applications. Typically, in academic datasets [19–21], the data for training and testing are usually sampled from similar distributions, from similar time scopes and corpora. However, in real-world applications, there is less control over the distribution of data during inference. This can cause knowledge conflicts between training and testing data, particularly in scenarios where knowledge acquisition is used to extract knowledge from new publications, where finding novel knowledge is more important than extracting known ones. Moreover, testing data may come from different corpora or semantic classes from those used for training in real-world applications, making the trained knowledge acquisition systems less applicable and requiring higher robustness to distribution shifts. These discrepancies highlight the need for knowledge acquisition systems to be more robust in real-world applications. However, most existing studies on knowledge acquisition do not take into account these discrepancies, raising concerns about their effectiveness in real-world settings. In

addition, since the data in real-world scenarios are often obtained by crowdsourcing, they may have noisy labels that degrade the system's performance. Therefore, enhancing the system's robustness to noisy training labels is also a crucial problem.

Furthermore, building data-efficient knowledge acquisition systems is a challenge, especially for high-stake domains such as biomedicine and finance. Annotators for these domain-specific tasks require domain expertise, leading to the high cost of manual data annotation. As a result, most existing efforts focus on distant supervision [22, 23], which consists of linking entities to an existing knowledge base and using the relation in the knowledge base to annotate the text. However, distant supervision is not applicable when relations of our interest do not exist in the knowledge bases. Besides, it can suffer from the knowledge conflict problem mentioned earlier, since the knowledge base may lack current knowledge. Another line of work focuses on few-shot knowledge acquisition [24, 25]. This process consists of training a knowledge acquisition model on (abundant) data from other relations first, and then adapting it to relations of our interest using a few labels. However, such methods are not suitable for high-stakes domains where data for other relations is also scarce. Therefore, constructing data-efficient knowledge acquisition systems remains a crucial research problem.

In this dissertation, we aim to build robust and generalizable knowledge acquisition systems that are: (i) able to extract knowledge from context beyond individual sentences; (ii) robust to the shifts between training and testing data, and noise in the training data; and (iii) requiring minimal human efforts.

1.2 Challenges

In this dissertation, we address several key challenges. First, to extract knowledge from long contexts, the knowledge acquisition system must be able to identify salient contextual information between pairs of entities. In addition, since a pair of entities may have multiple relations, multi-label prediction also represents a challenge. Second, since current datasets for knowledge acquisition do not address the issue of robustness, we need to develop experimental settings to measure the robustness and dedicated frameworks to improve it. Third, to enhance data efficiency, we need to address the problem from two angles: (1) obtaining annotated data with minimal human effort and (2) utilizing labels more effectively through dedicated frameworks. The details of the challenges we address in this dissertation are listed below:

- **Knowledge acquisition from long contexts.** To efficiently and effectively extract knowledge from long contexts, the system needs to address two challenges:

  (i) *Identifying salient context.* Current datasets for long-context knowledge acquisition can contain up to thousands of words [26], whereas a single knowledge triple is usually expressed within only a few sentences. Therefore, it is important for systems to identify relevant context in order to avoid being overwhelmed by large amounts of irrelevant information. However, it is usually hard to annotate salient context by humans, and it can be challenging to learn how to locate relevant context from scratch. To address this issue, systems may need to leverage additional syntactic and semantic information in order to accurately identify relevant context.

(ii) *Multi-label prediction.* Since an entity pair may appear multiple times in context, it often has more than one relation types, making knowledge acquisition from long context a multi-label prediction problem. For example, in the DocRED dataset [18], over 15% of entity pairs have more than one relation type. Although the common practice is to convert this problem into several binary classification problems, it involves adjusting various thresholds for each relation type on development data, which may not generalize to the test data. Additionally, this approach encounters a calibration issue as different entity pairs may have different optimal thresholds.

- **Robustness of knowledge acquisition systems.** As mentioned earlier, robustness in this context comprises several dimensions, such as distributional shifts, knowledge conflicts, and noisy training labels. However, these aspects have been less explored in previous research, which calls for setting up appropriate experimental settings and metrics to evaluate these aspects. Another challenge in improving robustness lies in the fact that it is impossible to know the distribution of testing data before deploying a system. As a result, it is essential to develop strategies that can improve robustness in an unsupervised way.

- **Data-efficient knowledge acquisition.** In order to enhance data efficiency, there are two primary challenges that must be addressed. First, data annotation is a necessary step, unless distant supervision can be utilized. While for high-stake domains, data annotation is both time-consuming and highly expensive. As a result, a major challenge is how to decrease the cost of data annotation. Second, since unlabeled data is often abundant, another challenge is how to effectively leverage this data to improve the system.

1.3 Thesis Contributions

- We propose a framework for document-level relation extraction that addresses both the salient context and the multi-label prediction challenge. Specifically, we introduce localized context pooling, which leverages attention in PLMs to identify relevant context for entity pairs. Additionally, we propose the use of an adaptive-thresholding loss, which allows the model to learn an adaptive threshold is adaptive to entity pairs.

- We present a co-regularization framework that can efficiently train supervised models using noisy datasets, which consists of multiple models with different initialization and a regularization loss that enforces the outputs of different models to be similar. We also discuss various design strategies and balance efficiency with effectiveness.

- We propose to identify and reject instances that are from different distributions to training, making the model abstain from making unconfident predictions. To achieve this, we propose a framework for unsupervised out-of-distribution detection using contrastive learning to expand the boundaries between different classes. We also explore various combinations of contrastive learning losses and out-of-distribution scoring functions to achieve the best performance in this challenging task.

- We study the knowledge conflict problem in knowledge acquisition, where the system needs to extract the knowledge faithfully to the context and avoid memorizing knowledge in the

pretraining corpus. we focus on large language models and present two techniques, namely opinion-based prompts and counterfactual demonstrations.

- We propose to reduce the annotation cost using labeling rules. To solve the low coverage problem of rules, we propose a framework that learns to generalize labeling rules to semantically similar instances by neural rule grounding.

- To leverage unlabeled data in knowledge acquisition, we propose to first pretrain the PLMs based on matching-the-blanks objectives, and then finetune PLMs on task-specific data. To minimize the gap in training objectives between pretraining and finetuning, we propose to use a consistent contrastive loss in pretraining and finetuning, and use kNN-based inference. Furthermore, we design an MCCL finetuning objective, allowing one relation to form multiple different clusters, thus further reducing the distributional gap between pretraining and finetuning.

1.4 Thesis Overview

Part I: Knowledge Acquisition in Documents

- Chapter 2:

  This chapter introduces the ATLOP (**A**daptive **T**hresholding and **L**ocalized c**O**ntext **P**ooling) framework for document-level relation extraction. We propose two techniques, namely adaptive thresholding and localized context pooling, to address the challenges of multi-label classification and identifying salient context. Adaptive thresholding overcomes the limitations of a global threshold in prior work by introducing a learnable, entity-dependent threshold. Localized context pooling directly transfers the attention from PLMs to locate context that is salient for relation classification. Our experiments on three benchmark datasets for document-level RE, including DocRED in the general domain, as well as CDR and GDA in the biomedical domain, demonstrate the effectiveness of our proposed framework.

Part II: Robust Knowledge Acquisition

- Chapter 3:

  In this chapter, we introduce a co-regularization framework that addresses the challenge of noisy training labels. Our approach is motivated by studies showing that noisy labels are more difficult to memorize during training and are more prone to being forgotten than clean labels, making them identifiable during training. To mitigate this issue, we propose a simple co-regularization framework for information extraction. This framework consisits of several neural models with identical structures but different parameter initialization. These models are jointly optimized with task-specific losses and are regularized to produce similar predictions based on an agreement loss, which prevents overfitting on noisy labels. Our extensive experiments on two information extraction benchmarks demonstrate the effectiveness of our proposed framework.

- Chapter 4:

  In this chapter, we present our framework for detecting out-of-distribution (OOD) instances, which can lead to significant semantic shift problems during inference. In practice, it is crucial for a reliable model to identify such instances and either reject them during inference or pass them on to models that can handle them. To address this issue, we propose an unsupervised OOD detection method that uses only in-distribution (ID) data during training. Our approach involves fine-tuning PLMs with a contrastive loss, which improves the compactness of representations, making it easier to distinguish OOD instances from ID ones. We use the Mahalanobis distance in the model's penultimate layer to accurately detect OOD instances. Our comprehensive experiments demonstrate near-perfect OOD detection performance.

- Chapter 5:

  This chapter focuses on techniques for mitigating knowledge conflicts in large language models (LLMs). LLMs have demonstrated exceptional performance in knowledge-driven NLP tasks due to their ability to encode parametric knowledge about world facts. However, this reliance on parametric knowledge may cause them to miss contextual cues, leading to inaccurate predictions in context-sensitive NLP tasks. To address this issue, we propose techniques to assess and enhance LLMs' contextual faithfulness in two aspects: knowledge conflict and prediction with abstention. We show that LLMs' contextual faithfulness can be significantly improved using carefully designed prompting strategies, namely opinion-based prompts and counterfactual demonstrations. Opinion-based prompts reframe the context as a narrator's statement and inquire about the narrator's opinions, while counterfactual demonstrations use instances containing false facts to improve faithfulness in knowledge-conflict situations.

Part III: Data-efficient Knowledge Acquisition

- Chapter 6:

  In this chapter, we introduce the NERO framework, which enables efficient data annotation by creating labeling rules instead of seeking more instance-level labels from human annotators. These rules can be automatically mined from large text corpora and generalized using a soft rule-matching mechanism. Our framework consists of a relation extraction (RE) module and a soft matching module. The RE module can be instantiated with any neural model, while the soft matching module learns to match rules with semantically similar sentences. This process allows raw corpora to be automatically labeled. We conduct extensive experiments and analysis on two widely-used public datasets, demonstrating the effectiveness of the proposed NERO framework compared to both rule-based and semi-supervised methods. Through user studies, we find NERO yields a 9.5x speedup of annotation compared to human annotation.

- Chapter 7:

  In this chapter, we introduce continual contrastive finetuning for data-efficient knowledge acquisition. The approach involves self-supervised learning, where the relation embedding is pretrained using a RE-based objective and then finetuned on labeled data using a classification-based objective. To bridge the gap between pretraining and finetuning, we propose using consistent objectives of contrastive learning for both stages of training. We also propose

a multi-center contrastive loss that allows one relation to form multiple clusters in the representation space, which better aligns with pretraining. Our experiments on two document-level RE datasets, demonstrate the effectiveness of our approach.

Part IV: Conclusion

- Chapter 8:

  This chapter summarizes the contributions of the thesis and discusses several future directions.

Chapter 2

Document-level Relation Extraction

Document-level relation extraction (RE) poses new challenges compared to its sentence-level counterpart. One document commonly contains multiple entity pairs, and one entity pair occurs multiple times in the document associated with multiple possible relations. We propose two novel techniques, adaptive thresholding and localized context pooling, to solve the multi-label and multi-entity problems. The adaptive thresholding replaces the global threshold for multi-label classification in the prior work with a learnable entities-dependent threshold. The localized context pooling directly transfers attention from pre-trained language models to locate relevant context that is useful to decide the relation. We experiment on three document-level RE benchmark datasets: DocRED, a recently released large-scale RE dataset, and two datasets CDR and GDA in the biomedical domain. Our ATLOP (**A**daptive **T**hresholding and **L**ocalized c**O**ntext **P**ooling) model achieves an F1 score of 63.4, and also significantly outperforms existing models on both CDR and GDA[1].

2.1 Introduction

Relation extraction (RE) aims to identify the relationship between two entities in a given text and plays an important role in information extraction. Existing work mainly focuses on sentence-level relation extraction, i.e., predicting the relationship between entities in a single sentence [28–30]. However, large amounts of relationships, such as relational facts from Wikipedia articles and biomedical literature, are expressed by multiple sentences in real-world applications [18, 31]. This problem, commonly referred to as document-level relation extraction, necessitates models that can capture complex interactions among entities in the whole document.

Compared to sentence-level RE, document-level RE poses unique challenges. For sentence-level RE datasets such as TACRED [20] and SemEval 2010 Task 8 [19], a sentence only contains one entity pair to classify. On the other hand, for document-level RE, one document contains multiple entity pairs, and we need to classify the relations of them all at once. It requires the RE model to identify and focus on the part of the document with relevant context for a particular entity pair. In addition, one entity pair can occur many times in the document associated with distinct relations for document-level RE, in contrast to one relation per entity pair for sentence-level RE. This multi-entity (multiple entity pairs to classify in a document) and multi-label (multiple relation types for a particular entity pair) properties of document-level relation extraction make it harder

[1]This chapter is based on Zhou, Huang, Ma & Huang [27].

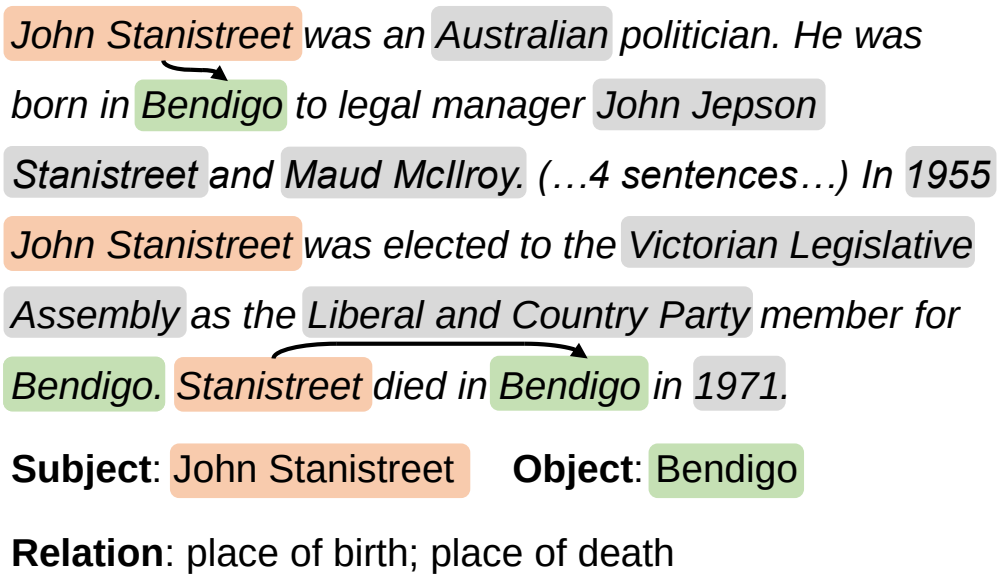

Figure 2.1: An example of multi-entity and multi-label problems from the DocRED dataset. Subject entity *John Stanistreet* (in orange) and object entity *Bendigo* (in green) express relations *place of birth* and *place of death*. The related entity mentions are connected by lines. Other entities in the document are highlighted in grey.

than its sentence-level counterpart. Figure 2.1 shows an example from the DocRED dataset [18]. The task is to classify the relation types of pairs of entities (highlighted in color). For a particular entity pair (*John Stanistreet*, *Bendigo*), it expresses two relations *place of birth* and *place of death* by the first two sentences and the last sentence. Other sentences contain irrelevant information to this entity pair.

To tackle the multi-entity problem, most current approaches construct a document graph with dependency structures, heuristics, or structured attention [32–35], and then perform inference with graph neural models [36, 37]. The constructed graphs bridge entities that spread far apart in the document and thus alleviate the deficiency of RNN-based encoders [38, 39] in capturing long-distance information [40]. However, as transformer-based models [41] can implicitly model long-distance dependencies [42, 43], it is unclear whether graph structures still help on top of pre-trained language models such as BERT [14]. There have also been approaches to directly apply pre-trained language models without introducing graph structures [44, 45]. They simply average the embedding of entity tokens to obtain the entity embeddings and feed them into the classifier to get relation labels. However, each entity has the same representation in different entity pairs, which can bring noise from irrelevant context.

Instead of introducing graph structures, we propose a localized context pooling technique. This technique solves the problem of using the same entity embedding for all entity pairs. It enhances the entity embedding with additional context that is relevant to the current entity pair. Instead of training a new context attention layer from scratch, we directly transfer the attention heads from pre-trained language models to get entity-level attention. Then, for two entities in a pair, we merge their attentions by multiplication to find the context that is important to both of them.

For the multi-label problem, existing approaches reduce it to a binary classification problem. After training, a global threshold is applied to the class probabilities to get relation labels. This method involves heuristic threshold tuning and introduces decision errors when the tuned threshold from development data may not be optimal for all instances.

We propose the adaptive thresholding technique, which replaces the global threshold with a learnable threshold class. The threshold class is learned with our adaptive-threshold loss, which is a *rank-based* loss that pushes the logits of positive classes above the threshold and pulls the logits of negative classes below in model training. At the test time, we return classes that have higher logits than the threshold class as the predicted labels or return NA if such class does not exist. This technique eliminates the need for threshold tuning, and also makes the threshold adjustable to different entity pairs, which leads to much better results.

By combining the proposed two techniques, we propose a simple yet effective relation extraction model, named ATLOP (**A**daptive **T**hresholding and **L**ocalized c**O**ntext **P**ooling), to fully utilize the power of pre-trained language models [14, 46]. This model tackles the multi-label and multi-entity problems in document-level RE. Experiments on three document-level relation extraction datasets, DocRED [18], CDR [47], and GDA [48], demonstrate that our ATLOP model significantly outperforms the state-of-the-art methods. The contributions of our work are summarized as follows:

- We propose adaptive-thresholding loss, which enables the learning of an adaptive threshold that is dependent on entity pairs and reduces the decision errors caused by using a global threshold.
- We propose localized context pooling, which transfers pre-trained attention to grab related context for entity pairs to get better entity representations.
- We conduct experiments on three public document-level relation extraction datasets. Experimental results demonstrate the effectiveness of our ATLOP model that achieves state-of-the-art performance on three benchmark datasets.

2.2 Problem Formulation

Given a document d and a set of entities $\{e_i\}_{i=1}^{n}$, the task of document-level relation extraction is to predict a subset of relations from $\mathscr{R} \cup \{\text{NA}\}$ between the entity pairs $(e_s, e_o)_{s,o=1...n;s\neq o}$, where $\mathscr{R}$ is a pre-defined set of relations of interest, e_s, e_o are identified as subject and object entities, respectively. An entity e_i can occur multiple times in the document by entity mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$. A relation exists between entities (e_s, e_o) if it is expressed by any pair of their mentions. The entity pairs that do not express any relation are labeled NA. At the test time, the model needs to predict the labels of all entity pairs $(e_s, e_o)_{s,o=1...n;s\neq o}$ in document d.

2.3 Enhanced BERT Baseline

In this section, we present our base model for document-level relation extraction. We build our model based on existing BERT baselines [18, 44] and integrate other techniques to further improve the performance.

2.3.1 Encoder

Given a document $d = [x_t]_{t=1}^{l}$, we mark the position of entity mentions by inserting a special symbol "*" at the start and end of mentions. It is adapted from the entity marker technique [16, 20, 49]. We then feed the document into a pre-trained language model to obtain the contextual embeddings:

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_l] = \text{BERT}([x_1, x_2, ..., x_l]). \tag{2.1}$$

Following previous work [31, 50], the document is encoded once by the encoder, and the classification of all entity pairs is based on the same contextual embedding. We take the embedding of "*" at the start of mentions as the mention embeddings. For an entity e_i with mentions $\{m_j^i\}_{j=1}^{N_{e_i}}$, we apply logsumexp pooling [51], a smooth version of max pooling, to get the entity embedding $\mathbf{h}_{e_i}$.

$$\mathbf{h}_{e_i} = \log \sum_{j=1}^{N_{e_i}} \exp\left(\mathbf{h}_{m_j^i}\right). \tag{2.2}$$

This pooling accumulates signals from mentions in the document. It shows better performance compared to mean pooling in experiments.

2.3.2 Binary Classifier

Given the embedding $(\mathbf{h}_{e_s}, \mathbf{h}_{e_o})$ of an entity pair e_s, e_o computed by equation 2.2, we map the entities to hidden states $\mathbf{z}$ with a linear layer followed by non-linear activation, then calculate the probability of relation r by bilinear function and sigmoid activation. This process is formulated as:

$$\mathbf{z}_s = \tanh\left(\mathbf{W}_s \mathbf{h}_{e_s}\right), \tag{2.3}$$

$$\mathbf{z}_o = \tanh\left(\mathbf{W}_o \mathbf{h}_{e_o}\right), \tag{2.4}$$

$$\text{P}\left(r|e_s, e_o\right) = \sigma\left(\mathbf{z}_s^\intercal \mathbf{W}_r \mathbf{z}_o + b_r\right),$$

where $\mathbf{W}_s \in \mathbb{R}^{d\times d}$, $\mathbf{W}_o \in \mathbb{R}^{d\times d}$, $\mathbf{W}_r \in \mathbb{R}^{d\times d}, b_r \in \mathbb{R}$ are model parameters. The representation of one entity is the same among different entity pairs. To reduce the number of parameters in the bilinear classifier, we use the group bilinear [52, 53], which splits the embedding dimensions into k equal-sized groups and applies bilinear within the groups:

$$\begin{aligned}
\left[\mathbf{z}_s^1; ...; \mathbf{z}_s^k\right] &= \mathbf{z}_s, \\
\left[\mathbf{z}_o^1; ...; \mathbf{z}_o^k\right] &= \mathbf{z}_o, \\
\text{P}\left(r|e_s, e_o\right) &= \sigma\left(\sum_{i=1}^{k} \mathbf{z}_s^{i\intercal} \mathbf{W}_r^i \mathbf{z}_o^i + b_r\right),
\end{aligned} \tag{2.5}$$

where $\mathbf{W}_r^i \in \mathbb{R}^{d/k\times d/k}$ for $i = 1...k$ are model parameters, $\text{P}\left(r|e_s, e_o\right)$ is the probability that relation r is associated with the entity pair (e_s, e_o). In this way, we can reduce the number of parameters from d^2 to d^2/k. We use the binary cross entropy loss for training. During inference, we tune a

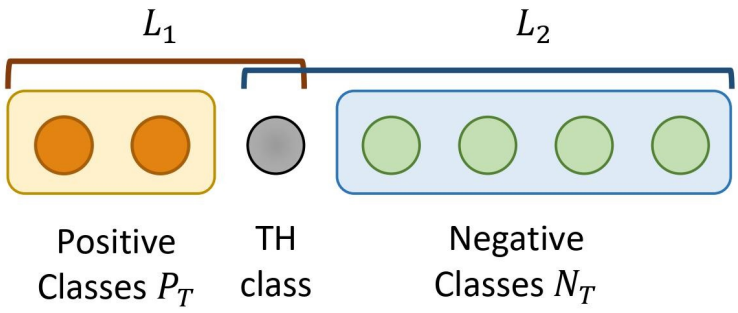

Figure 2.2: An artificial illustration of our proposed adaptive-thresholding loss. A TH class is introduced to separate positive classes and negative classes: positive classes would have higher probabilities than TH, and negative classes would have lower probabilities than TH.

global threshold θ that maximizes evaluation metrics (F_1 score for RE) on the development set and return r as an associated relation if $\mathrm{P}(r|e_s,e_o) > \theta$ or return NA if no relation exists.

Our enhanced base model achieves near state-of-the-art performance in our experiments, significantly outperforms existing BERT baselines.

2.4 Adaptive Thresholding

The RE classifier outputs the probability $\mathrm{P}(r|e_s,e_o)$ within the range $[0,1]$, which needs thresholding to be converted to relation labels. As the threshold neither has a closed-form solution nor is differentiable, a common practice for deciding threshold is enumerating several values in the range $(0,1)$ and picking the one that maximizes the evaluation metrics (F_1 score for RE). However, the model may have different confidence for different entity pairs or classes in which one global threshold does not suffice. The number of relations varies (multi-label problem) and the models may not be globally calibrated so that the same probability does not mean the same for all entity pairs. This problem motivates us to replace the global threshold with a learnable, adaptive one, which can reduce decision errors during inference.

For the convenience of explanation, we split the labels of entity pair $T = (e_s,e_o)$ into two subsets: positive classes $\mathscr{P}_T$ and negative classes $\mathscr{N}_T$, which are defined as follows:

- positive classes $\mathscr{P}_T \subseteq \mathscr{R}$ are the relations that exist between the entities in T. If T does not express any relation, $\mathscr{P}_T$ is empty.

- negative classes $\mathscr{N}_T \subseteq \mathscr{R}$ are the relations that do not exist between the entities. If T does not express any relation, $\mathscr{N}_T = \mathscr{R}$.

If an entity pair is classified correctly, the logits of positive classes should be higher than the threshold while those of negative classes should be lower. Here we introduce a threshold class TH, which is automatically learned in the same way as other classes (see Eq.2.5). At the test time, we return classes with higher logits than the TH class as positive classes or return NA if such classes do not exist. This threshold class learns an entities-dependent threshold value. It is a substitute for the global threshold and thus eliminates the need for tuning threshold on the development set.

To learn the new model, we need a special loss function that considers the TH class. We design our adaptive-thresholding loss based on the standard categorical cross-entropy loss. The loss function is broken down into two parts as shown below:

$$\mathscr{L}_1 = -\sum_{r \in \mathscr{P}_T} \log\left(\frac{\exp(\text{logit}_r)}{\sum_{r' \in \mathscr{P}_T \cup \{\text{TH}\}} \exp(\text{logit}_{r'})}\right),$$
$$\mathscr{L}_2 = -\log\left(\frac{\exp(\text{logit}_{\text{TH}})}{\sum_{r' \in \mathscr{N}_T \cup \{\text{TH}\}} \exp(\text{logit}_{r'})}\right),$$
$$\mathscr{L} = \mathscr{L}_1 + \mathscr{L}_2.$$

The first part $\mathscr{L}_1$ involves positive classes and the TH class. Since there may be multiple positive classes, the total loss is calculated as the sum of categorical cross entropy losses on all positive classes [54, 55]. $\mathscr{L}_1$ pushes the logits of all positive classes to be higher than the TH class. It is not used if there is no positive label. The second part $\mathscr{L}_2$ involves the negative classes and threshold class. It is a categorical cross entropy loss with TH class being the true label. It pulls the logits of negative classes to be lower than the TH class. Two parts are simply summed for the total loss.

The proposed adaptive-thresholding loss is illustrated in Figure 2.2. It obtains a large performance gain to the global threshold in our experiments.

2.5 Localized Context Pooling

The logsumexp pooling (see Eq. 2.2) accumulates the embedding of all mentions for an entity across the whole document and generates one embedding for this entity. The entity embedding from this document-level global pooling is then used in the classification of all entity pairs. However, for an entity pair, some context of the entities may not be relevant. For example, in Figure 2.1, the second mention of *John Stanistreet* and its context are irrelevant to the entity pair (*John Stanistreet*, *Bendigo*). Therefore, it is better to have a localized representation that only attends to the relevant context in the document that is useful to decide the relation for this entity pair.

Therefore we propose localized context pooling, where we enhance the embedding of an entity pair with an additional local context embedding that is related to both entities. In this work, since we use pre-trained transformer-based models as the encoder, which has already learned token-level dependencies by multi-head self-attention [41], we consider directly using their attention heads for localized context pooling. This method transfers the well-learned dependencies from the pre-trained language model without learning new attention layers from scratch.

Specifically, given a pre-trained multi-head attention matrix $\mathbf{A} \in \mathbb{R}^{H \times l \times l}$, where $\mathbf{A}_{ijk}$ represents attention from token j to token k in the i^{th} attention head, we first take the attention from the "*" symbol as the mention-level attention, then average the attention over mentions of the same entity to obtain entity-level attention $\mathbf{A}^E_i \in \mathbb{R}^{H \times l}$, which denotes attention from the i^{th} entity to all tokens.

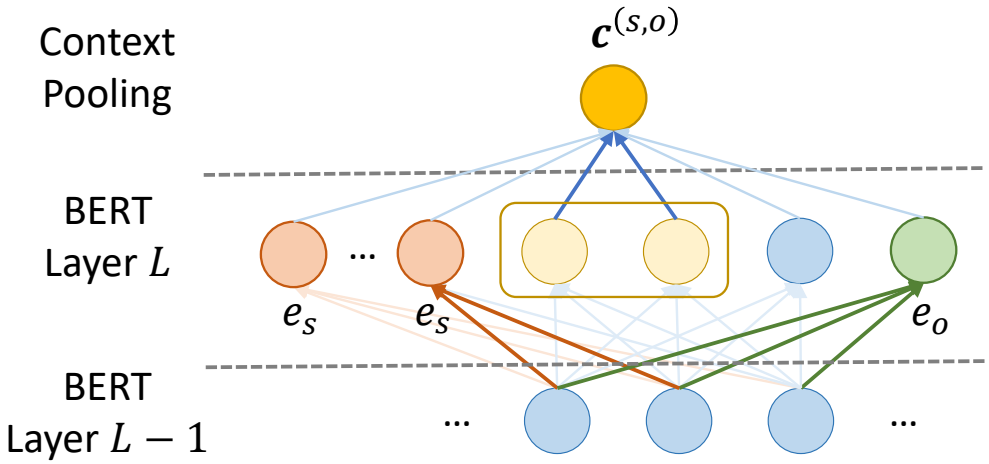

Figure 2.3: Illustration of localized context pooling. Tokens are weighted averaged to form the localized context $\mathbf{c}^{(s,o)}$ of the entity pair (e_s, e_o). The weights of tokens are derived by multiplying the attention weights of the subject entity e_s and the object entity e_o from the last transformer layer so that only the tokens that are important to both entities (highlighted in light yellow) receive higher weights.

Then given an entity pair (e_s, e_o), we locate the local context that is important to both e_s and e_o by multiplying their entity-level attention, and obtain the localized context embedding $\mathbf{c}^{(s,o)}$ by:

$$\begin{aligned}
\mathbf{A}^{(s,o)} &= \mathbf{A}_s^E \cdot \mathbf{A}_o^E, \\
\mathbf{q}^{(s,o)} &= \sum_{i=1}^{H} \mathbf{A}_i^{(s,o)}, \\
\mathbf{a}^{(s,o)} &= \mathbf{q}^{(s,o)} / \mathbf{1}^\intercal \mathbf{q}^{(s,o)}, \\
\mathbf{c}^{(s,o)} &= \mathbf{H}^\intercal \mathbf{a}^{(s,o)},
\end{aligned}$$

where $\mathbf{H}$ is the contextual embedding in Eq. 2.1. The localized context embedding is then fused into the globally pooled entity embedding to obtain entity representations that are different for different entity pairs, by modifying the original linear layer in Eq. 2.3 and Eq. 2.4 as follows:

$$\mathbf{z}_s^{(s,o)} = \tanh\left(\mathbf{W}_s \mathbf{h}_{e_s} + \mathbf{W}_{c_1} \mathbf{c}^{(s,o)}\right), \tag{2.6}$$

$$\mathbf{z}_o^{(s,o)} = \tanh\left(\mathbf{W}_o \mathbf{h}_{e_o} + \mathbf{W}_{c_2} \mathbf{c}^{(s,o)}\right), \tag{2.7}$$

where $\mathbf{W}_{c_1}, \mathbf{W}_{c_2} \in \mathbb{R}^{d \times d}$ are model parameters. The proposed localized context pooling is illustrated in Figure 2.3. In experiments, we use the attention matrix from the last transformer layer.

Statistics	DocRED	CDR	GDA
# Train	3053	500	23353
# Dev	1000	500	5839
# Test	1000	500	1000
# Relations	97	2	2
Avg.# entities per Doc.	19.5	7.6	5.4

Table 2.1: Statistics of the datasets in experiments.

Hyperparam	**DocRED**		**CDR**	**GDA**
	BERT	RoBERTa	SciBERT	SciBERT
Batch size	4	4	4	16
# Epoch	30	30	30	10
lr for encoder	5e-5	3e-5	2e-5	2e-5
lr for classifier	1e-4	1e-4	1e-4	1e-4

Table 2.2: Hyper-parameters in training.

2.6 Experiments

2.6.1 Datasets

We evaluate our ATLOP model on three public document-level relation extraction datasets. The dataset statistics are shown in Table 2.1.

- **DocRED** [18] is a large-scale crowdsourced dataset for document-level RE. It is constructed from Wikipedia articles. DocRED consists of 3053 documents for training. For entity pairs that express relation(s), about 7% of them have more than one relation label.
- **CDR** [47] is a human-annotated dataset in the biomedical domain. It consists of 500 documents for training. The task is to predict the binary interactions between Chemical and Disease concepts.
- **GDA** [48] is a large-scale dataset in the biomedical domain. It consists of 29192 articles for training. The task is to predict the binary interactions between Gene and Disease concepts. We follow Christopoulou, Miwa & Ananiadou [34] to split the training set into an 80/20 split as training and development sets.

2.6.2 Experiment Settings

Our model is implemented based on Huggingface's Transformers [58]. We use cased BERT-base [14] or RoBERTa-large [46] as the encoder on DocRED, and cased SciBERT [59] on CDR and GDA. We use mixed-precision training [60] based on the Apex library[2]. Our model is optimized

[2] https://github.com/NVIDIA/apex

Model	Dev		Test	
	Ign F_1	F_1	Ign F_1	F_1
Sequence-based Models				
CNN [18]	41.58	43.45	40.33	42.26
BiLSTM [18]	48.87	50.94	48.78	51.06
Graph-based Models				
BiLSTM-AGGCN [37]	46.29	52.47	48.89	51.45
BiLSTM-LSR [35]	48.82	55.17	52.15	54.18
BERT-LSR$_{\text{BASE}}$ [35]	52.43	59.00	56.97	59.05
Transformer-based Models				
BERT$_{\text{BASE}}$ [44]	-	54.16	-	53.20
BERT-TS$_{\text{BASE}}$ [44]	-	54.42	-	53.92
HIN-BERT$_{\text{BASE}}$ [45]	54.29	56.31	53.70	55.60
CorefBERT$_{\text{BASE}}$ [56]	55.32	57.51	54.54	56.96
CorefRoBERTa$_{\text{LARGE}}$ [56]	57.35	59.43	57.90	60.25
Our Methods				
BERT$_{\text{BASE}}$ (our implementation)	54.27 ± 0.28	56.39 ± 0.18	-	-
BERT-E$_{\text{BASE}}$	56.51 ± 0.16	58.52 ± 0.19	-	-
BERT-ATLOP$_{\text{BASE}}$	59.22 ± 0.15	61.09 ± 0.16	59.31	61.30
RoBERTa-ATLOP$_{\text{LARGE}}$	$\mathbf{61.32 \pm 0.14}$	$\mathbf{63.18 \pm 0.19}$	**61.39**	**63.40**

Table 2.3: Main results (%) on the development and test set of DocRED. We report the mean and standard deviation of F_1 on the development set by conducting 5 runs of training using different random seeds. We report the official test score of the best checkpoint on the development set.

Model	CDR	GDA
BRAN [31]	62.1	-
CNN [57]	62.3	-
EoG [34]	63.6	81.5
LSR [35]	64.8	82.2
SciBERT (our implementation)	65.1 ± 0.6	82.5 ± 0.3
SciBERT-E	65.9 ± 0.5	83.3 ± 0.3
SciBERT-ATLOP	$\mathbf{69.4 \pm 1.1}$	$\mathbf{83.9 \pm 0.2}$

Table 2.4: Test F_1 score (%) on CDR and GDA dataset. Our ATLOP model with the SciBERT encoder outperforms the current SOTA results.

with AdamW [61] using learning rates $\in \{2e{-}5, 3e{-}5, 5e{-}5, 1e{-}4\}$, with a linear warmup [62] for the first 6% steps followed by a linear decay to 0. We apply dropout [63] between layers with rate 0.1, and clip the gradients of model parameters to a max norm of 1.0. We perform early stopping based on the F_1 score on the development set. All hyper-parameters are tuned on the development set. We list some of the hyper-parameters in Table 2.2.

For models that use a global threshold, we search threshold values from $\{0.1, 0.2, ..., 0.9\}$ and pick the one that maximizes dev F_1. All models are trained with 1 Tesla V100 GPU. For the DocRED dataset, the training takes about 1 hour 45 minutes with BERT-base encoder and 3 hours 30 minutes with RoBERTa-large encoder. For CDR and GDA datasets, the training takes 20 minutes and 3 hours 30 minutes with SciBERT encoder, respectively.

2.6.3 Main Results

We compare ATLOP with sequence-based models, graph-based models, and transformer-based models on the DocRED dataset. The experiment results are shown in Table 2.3. Following Yao, Ye, Li, Han, Lin, Liu, *et al.* [18], we use F_1 and Ign F_1 in evaluation. The Ign F_1 denotes the F_1 score excluding the relational facts that are shared by the training and dev/test sets.

Sequence-based Models. These models use neural architectures such as CNN [64] and bidirectional LSTM [65] to encode the entire document, then obtain entity embeddings and predict relations for each entity pair with bilinear function.

Graph-based Models. These models construct document graphs by learning latent graph structures of the document and perform inference with graph convolutional network [66]. We include two state-of-the-art graph-based models, AGGCN [37] and LSR [35], for comparison. The result of AGGCN is from the re-implementation by Nan, Guo, Sekulic & Lu [35].

Transformer-based Models. These models directly adapt pre-trained language models to document-level RE without using graph structures. They can be further divided into pipeline models (BERT-TS [44]), hierarchical models (HIN-BERT [45]), and pre-training methods (CorefBERT and CorefRoBERTa [56]). We also include the BERT baseline [44] and our re-implemented BERT baseline in comparison.

We find that our re-implemented BERT baseline gets significantly better results than Wang, Focke, Sylvester, Mishra & Wang [44], and outperforms the state-of-the-art RNN-based model BiLSTM-LSR by 1.2%. It demonstrates that pre-trained language models can capture long-distance dependencies among entities without explicitly using graph structures. After integrating other techniques, our enhanced baseline BERT-E_{BASE} achieves an F1 score of 58.52%, which is close to the current state-of-the-art model BERT-LSR_{BASE}. Our BERT-$ATLOP_{BASE}$ model further improves the performance of BERT-E_{BASE} by 2.6%, demonstrating the efficacy of the proposed two novel techniques. Using RoBERTa-large as the encoder, our ALTOP model achieves an F1 score of 63.40%, which is a new state-of-the-art result on DocRED.

2.6.4 Results on Biomedical Datasets

Experiment results on two biomedical datasets are shown in Table 2.4. Verga, Strubell & McCallum [31] and Nguyen & Verspoor [57] are both sequence-based models that use self-attention network and CNN as the encoders, respectively. Christopoulou, Miwa & Ananiadou [34] and Nan, Guo, Sekulic & Lu [35] use graph-based models that construct document graphs by heuristics or structured attention, and perform inference with graph neural network. To our best knowledge, transformer-based pre-trained language models have not been applied to document-level RE datasets in the biomedical domain. In experiments, we replace the encoder with SciBERT, which is pre-trained on multi-domain corpora of scientific publications. The SciBERT baseline already outperforms all

Model	Ign F_1	F_1
BERT-ATLOP$_{\text{BASE}}$	59.22	61.09
– Adaptive Thresholding	58.32	60.20
– Localized Context Pooling	58.19	60.12
– Adaptive-Thresholding Loss	39.52	41.74
BERT-E$_{\text{BASE}}$	56.51	58.52
– Entity Marker	56.22	58.28
– Group Bilinear	55.51	57.54
– Logsumexp Pooling	55.35	57.40

Table 2.5: Ablation study of ATLOP on DocRED. We turn off different components of the model one at a time. These ablation results show that both adaptive thresholding and localized context pooling are effective. Logsumexp pooling and group bilinear both bring noticeable gain to the baseline.

Strategy	Dev F_1	Test F_1
Global Thresholding	60.14	60.62
Per-class Thresholding	**61.73**	60.35
Adaptive Thresholding	61.27	**61.30**

Table 2.6: Result of different thresholding strategies on DocRED. Our adaptive thresholding consistently outperforms other strategies on the test set.

existing methods. Our SciBERT-ATLOP model further improves the F_1 score by 4.3% and 1.4% on CDR and GDA, respectively, yielding new state-of-the-art results on these two datasets.

2.6.5 Ablation Study

To show the efficacy of our proposed techniques, we conduct two sets of ablation studies on ATLOP and enhanced baseline, by turning off one component at a time. We observe that all components contribute to model performance. The adaptive thresholding and localized context pooling are equally important to model performance, leading to a drop of 0.89% and 0.97% in dev F_1 score respectively when removed from ATLOP. Note that the adaptive thresholding only works when the model is optimized with the adaptive-thresholding loss. Applying adaptive thresholding to models trained with binary cross entropy results in dev F_1 of 41.74%.

For our enhanced baseline model BERT-E$_{\text{BASE}}$, both group bilinear and logsumexp pooling lead to about 1% increase in dev F_1. We find the improvement from entity markers is minor (0.24% in dev F_1) but still use the technique in the model as it makes the derivation of mention embedding and mention-level attention easier.

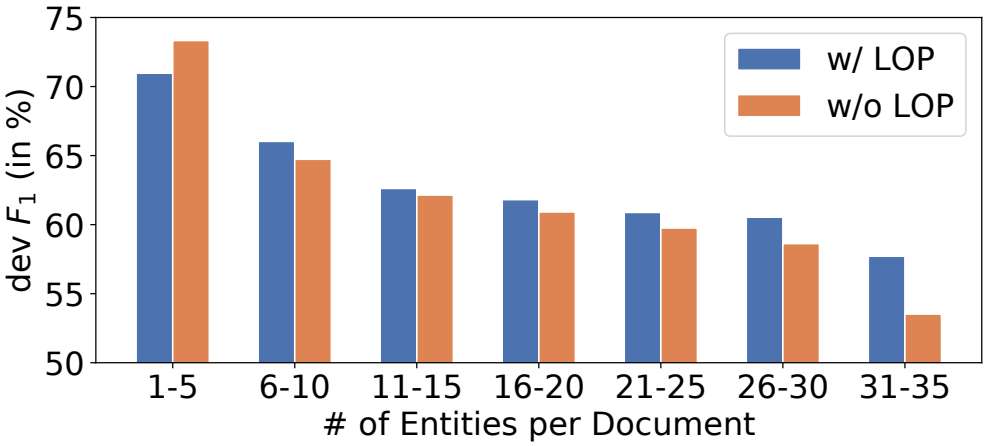

Figure 2.4: Dev F_1 score of documents with the different number of entities on DocRED. Our localized context pooling achieves better results when the number of entities is larger than 5. The improvement becomes more significant when the number of entities increases.

2.6.6 Analysis of Thresholding

Global thresholding does not consider the variations of model confidence in different classes or instances, and thus yields suboptimal performance. One interesting question is whether we can improve global thresholding by tuning different thresholds for different classes. To answer this question, We try to tune different thresholds on different classes to maximize the dev F_1 score on DocRED using the cyclic optimization algorithm [67]. Results are shown in Table 2.6. We find that using per-class thresholding significantly improves the dev F_1 score to 61.73%, which is even higher than the result of adaptive thresholding. However, this gain does not transfer to the test set. The result of per-class thresholding is even worse than global thresholding. It indicates that tuning per-class thresholding after training can lead to severe over-fitting to the development set. While our adaptive thresholding technique learns the threshold in training, which can generalize to the test set.

2.6.7 Analysis of Context Pooling

To show that our localized context pooling (LOP) technique mitigates the multi-entity issue, we divide the documents in the development set of DocRED into different groups by the number of entities, and evaluate models trained with or without localized context pooling on each group. Experiment results are shown in Figure 2.4. We observe that for both models, their performance gets worse when the document contains more entities. The model w/ LOP consistently outperforms the model w/o LOP except when the document contains very few entities (1 to 5), and the improvement gets larger when the number of entities increases. However, the number of documents that only contain 1 to 5 entities is very small (4 in the dev set), and the documents in DocRED contain 19 entities on average. Therefore our localized context pooling still improves the overall F_1 score significantly. This indicates that the localized context pooling technique can capture related context for entity pairs and thus alleviates the multi-entity problem.

We also visualize the context weights of the example in Figure 2.1. As shown in Figure 2.5, our localized context pooling gives high weights to *born* and *died*, which are most relevant to both

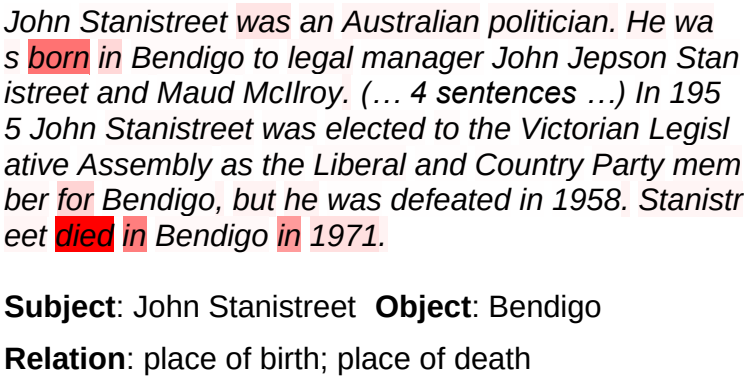

Figure 2.5: Context weights of an example from DocRED. We visualize the weight of context tokens $\mathbf{a}^{(s,o)}$ in localized context pooling. The model attends to the most relevant context *born* and *died* for entity pair (*John Stanistreet*, *Bendigo*).

entities (*John Stanistreet*, *Bendigo*). These two tokens are also evidence for the two ground truth relationships *place of birth* and *place of death*, respectively. Tokens like *elected* and *politician* get much smaller weights because they are only related to the subject entity *John Stanistreet*. The visualization demonstrates that the localized context can locate the context that is related to both entities.

2.7 Related Work

Early research efforts on relation extraction concentrate on predicting the relationship between two entities within a sentence. Various approaches including rule-based methods [68, 69], sequence-based methods [20, 28, 70], graph-based methods [29, 30, 37, 71], transformer-based methods [15, 16], and pre-training methods [49, 72] have been shown effective in tackling this problem.

However, as large amounts of relationships are expressed by multiple sentences [18, 31], recent work starts to explore document-level relation extraction. Most approaches on document-level RE are based on document graphs, which were introduced by Quirk & Poon [73]. Specifically, they use words as nodes and inner and inter-sentential dependencies (dependency structures, coreferences, etc.) as edges. This document graph provides a unified way of extracting the features for entity pairs. Later work extends the idea by improving neural architectures [31, 32, 51, 74, 75] or adding more types of edges [34, 35, 76]. In particular, Christopoulou, Miwa & Ananiadou [34] constructs nodes of different granularities (sentence, mention, entity), connects them with heuristically generated edges, and infers the relations with an edge-oriented model [77]. Nan, Guo, Sekulic & Lu [35] treats the document graph as a latent variable and induces it by structured attention [33]. This work also proposes a refinement mechanism to enable multi-hop information aggregation from the whole document. Their LSR model achieved state-of-the-art performance on document-level RE.

There have also been models that directly apply pre-trained language models without introducing document graphs, since edges such as dependency structures and coreferences can be automatically learned by pre-trained language models [42, 43, 78, 79]. In particular, Wang, Focke, Sylvester, Mishra & Wang [44] proposes a pipeline model that first predicts whether a relationship exists in an entity pair and then predicts the specific relation types. Tang, Cao, Zhang, Cao, Fang, Wang,

et al. [45] proposes a hierarchical model that aggregates entity information from the entity level, sentence level, and document level. Ye, Lin, Du, Liu, Sun & Liu [56] introduces a copy-based training objective to pre-training, which enhances the model's ability in capturing coreferential information and brings noticeable gain on various NLP tasks that require coreferential reasoning.

However, none of the models focus on the multi-entity and multi-label problems, which are among the key differences of document-level RE to its sentence-level RE counterpart. Our ATLOP model deals with the two problems by two novel techniques: adaptive thresholding and localized context pooling, and significantly outperforms existing models.

2.8 Conclusion

In this work, we propose the ATLOP model for document-level relation extraction, which features two novel techniques: adaptive thresholding and localized context pooling. The adaptive thresholding technique replaces the global threshold in multi-label classification with a learnable threshold class that can decide the best threshold for each entity pair. The localized context pooling utilizes pre-trained attention heads to locate relevant context for entity pairs and thus helps in alleviating the multi-entity problem. Experiments on three public document-level relation extraction datasets demonstrate that our ATLOP model significantly outperforms existing models and yields new state-of-the-art results on all datasets.

Chapter 3

Noisy Label Learning

Recent information extraction approaches have relied on training deep neural models. However, such models can easily overfit noisy labels and suffer from performance degradation. While it is very costly to filter noisy labels in large learning resources, recent studies show that such labels take more training steps to be memorized and are more frequently forgotten than clean labels, therefore are identifiable in training. Motivated by such properties, we propose a simple co-regularization framework for entity-centric information extraction, which consists of several neural models with identical structures but different parameter initialization. These models are jointly optimized with the task-specific losses and are regularized to generate similar predictions based on an agreement loss, which prevents overfitting on noisy labels. Extensive experiments on two widely used but noisy benchmarks for information extraction, TACRED and CoNLL03, demonstrate the effectiveness of our framework[1].

3.1 Introduction

Deep neural models have achieved significant success on various information extraction (IE) tasks. However, when training labels contain noise, deep neural models can easily overfit the noisy labels, leading to severe performance degradation [81, 82]. Unfortunately, labeling on large corpora, regardless of using human annotation [83] or automated heuristics [84], inevitably suffers from labeling errors. This problem has even drastically affected widely used benchmarks, such as CoNLL03 [85] and TACRED [20], where a notable portion of incorrect labels have been caused in annotation and largely hindered the performance of SOTA systems [86, 87]. Hence, developing a robust learning method that better tolerates noisy supervision represents an urged challenge for emerging IE models.

So far, few research efforts have been made to developing noise-robust IE models, and existing work mainly focuses on the weakly supervised or distantly supervised setting [88–91]. Most of such methods typically depend on multi-instance learning that relies on bags of instances provided by distant supervision [14, 23, 88] or require an additional clean and sufficiently large reference dataset to develop a noise filtering model [92]. Accordingly, those methods may not be generally adapted to supervised training settings, where the aforementioned auxiliary learning resources are not always available. Particularly, CrossWeigh [93] is a representative work that denoises a natural language dataset without using extra learning resources. This method trains multiple independent models

[1]This chapter is based on Zhou & Chen [80].

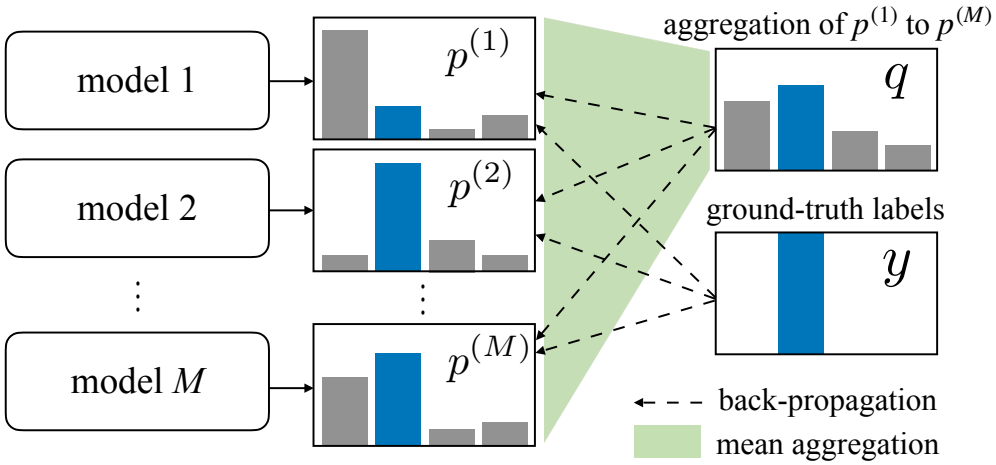

Figure 3.1: Illustration of our co-regularization framework. The base models are jointly optimized with the task-specific loss from label y and an agreement loss, which regularizes the models to generate similar predictions to the aggregated soft target probability $\boldsymbol{q}$.

on different partitions of training data and downweighs instances on which the models disagree. Though effective, a method of this kind requires training tens of redundant neural models, leading to excessive computational overhead for large models. As far as we know, the problem of noisy labels in supervised learning for IE tasks has not been well investigated.

We aim to develop a general denoising framework that can easily incorporate existing supervised learning models for entity-centric IE tasks. Our method is motivated by studies [81, 94] showing that noisy labels often have delayed learning curves, as incorrectly labeled instances are more likely to contradict the inductive bias captured by the model. Hence, noisy label instances take a longer time to be picked up by neural models and are frequently forgotten in later epochs. Therefore, predictions by more than one model tend to disagree on such instances. Accordingly, we propose a simple yet effective co-regularization framework to handle noisy training labels, as illustrated in Figure 3.1. Our framework consists of two or more neural classifiers with identical structures but different initialization. In training, all classifiers are optimized on the training data with the task-specific loss and jointly regularized with regard to an agreement loss that is defined as the Kullback–Leibler (KL) divergence among predicted probability distributions. Then for instances where a classifier's predictions disagree with labels, the agreement loss encourages the classifier to give similar predictions to the other classifier(s) instead of the actual (possibly noisy) labels. In this way, the framework prevents the incorporated classifiers from overfitting noisy labels.

We apply the framework to two important entity-centric IE tasks, named entity recognition (NER) and relation extraction (RE). We conduct extensive experiments on two prevalent but noisy benchmarks, CoNLL03 for NER and TACRED for RE, and apply the proposed learning frameworks to train various models from prior studies for these two tasks. The results demonstrate the effectiveness of our method in noise-robust training, leading to promising and consistent performance improvement. We present contributions as follows:

- We propose a general co-regularization framework that can effectively learn supervised IE models from noisy datasets without the need for any extra learning resources.

- We discuss in detail the different design strategies of the framework and the trade-off between efficiency and effectiveness.

- Extensive experiments on NER and RE demonstrate that our framework yields promising improvements on various SOTA models and outperforms existing denoising frameworks.

3.2 Method

We focus on developing a noise-robust learning framework that improves supervised models for entity-centric IE tasks. In such tasks, (noisy) labels can be assigned to either individual tokens (NER) or pairs of entities (RE) in natural language text. Specifically, $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ is a noisily labeled dataset, where each data instance consists of a lexical sequence or a context $\boldsymbol{x}$, and a label y. y is annotated either on tokens of $\boldsymbol{x}$ for NER or on a pair of entity mentions in $\boldsymbol{x}$ for RE. For some instances in $\mathscr{D}$, the labels are incorrect. Our objective is to learn a noise-robust model f with the presence of such noisily labeled instances from $\mathscr{D}$ without using external resources such as a clean development dataset [92].

3.2.1 Learning Process

Our framework is motivated by the delayed learning curve of a neural model on noisy data, compared with learning on clean data. On noisy data, neural models tend to fit easy and clean instances that are more consistent with the well-represented patterns of data in early steps but need more steps to capture noise [81]. Moreover, learned noisy examples tend to be frequently forgotten in later epochs [94] since they conflict with the general inductive bias represented by the clean data majority. Therefore, model prediction is likely to be consistent with the clean labels while is often inconsistent or oscillates on noisy labels over different training epochs. As a result, labels that are different from the model's predictions in the later epochs of training are likely to be noisy and should be down-weighted or rectified so as to reduce their impact on optimization.

The proposed framework incorporates several copies of a task-specific IE model with the same architecture but different (random) parameter initialization. These IE models are jointly optimized on the noisy dataset based on their task-specific losses as well as on an agreement loss. During training, the predicted probability distributions from models are aggregated as a soft target probability, which represents the models' estimations of the true label. The agreement loss is responsible for encouraging these models to generate similar predictions to the soft target probability. In this learning process, models starting their training from varied initialization generate different decision boundaries. By aggregating their predictions, the soft target probability can better separate noisy labels from clean labels that have not yet been learned.

The learning process of our framework is described in Algorithm 1. It consists of M ($M \geq 2$) copies of the task-specific model, denoted $\{f_k\}_{k=1}^{M}$, with different initialization. Regarding initialization, for models that are trained from scratch, all parameters are randomly initialized. Otherwise, for those that are built upon pre-trained language models, only the parameters that are external to the language models (*e.g.*, those of a downstream softmax classifier) are randomly initialized, while the pre-trained parameters are the same. Once initialized, our framework trains those models in two phases. The first $\alpha\%$ training steps undergo a *warm-up* phase, where α is a

Algorithm 1: Learning Process

Input: Dataset $\mathscr{D}$, hyperparameters T, α, M, γ.
Output: A trained model f.
Initialize M neural models $\{f_k\}_{k=1}^{M}$.
for $t = 1...T$ **do**
  Sample a batch $\mathscr{B}$ from $\mathscr{D}$.
  Calculate task losses $\{\mathscr{L}_{\text{sup}}^{(k)}\}_{k=1}^{M}$.
  $\mathscr{L}_T = \frac{1}{M}\sum_{k=1}^{M}\mathscr{L}_{\text{sup}}^{(k)}$
  if $t < \alpha\% \times T$ **then** // Warmup
    Update model parameters w.r.t. $\mathscr{L}_T$.
  else
    Get the probability distribution of classes $\{\boldsymbol{p}\}_{k=1}^{M}$ with M models.
    Calculate the soft target probability $\boldsymbol{q}$ by Equation 3.1.
    Calculate the agreement loss $\mathscr{L}_{\text{agg}}$ by Equation 3.2 and Equation 3.3.
    $\mathscr{L} = \mathscr{L}_T + \gamma \cdot \mathscr{L}_{\text{agg}}$.
    Update model parameters w.r.t. $\mathscr{L}$.
Return f_1 or the best-performing model.

hyperparameter. This phase seeks to help the model reach initial convergence on the task. When a new batch comes in, we first calculate the task-specific training losses on M models $\{\mathscr{L}_{\text{sup}}^{(k)}\}_{k=1}^{M}$ and average them as $\mathscr{L}_T$, then update model parameters w.r.t. $\mathscr{L}_T$. After the warm-up phase, an agreement loss $\mathscr{L}_{\text{agg}}$ is further introduced to measure the distance from the predictions of M models to the soft target probability $\boldsymbol{q}$. Parameters are accordingly updated based on the joint loss $\mathscr{L}$, encouraging the model to generate predictions that are consistent with both the training labels and the soft target probability. The formalization of the loss function is described next (Section 3.2.2). In the end, we can either use the model f_1 or select the best-performing model for inference.

3.2.2 Co-regularization Objective

In our framework, the influence of noisy labels in training is decreased by optimizing the agreement loss. Specifically, given a batch of data instances $\mathscr{B} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, we first feed the instances to M incorporated models to get their predictions $\left\{\{\boldsymbol{p}_i^{(k)}\}_{i=1}^{N}\right\}_{k=1}^{M}$ on $\mathscr{B}$, where $\boldsymbol{p} \in \mathbb{R}^C$ is the predicted probability distribution of C classes. Then we calculate the soft target probability $\boldsymbol{q}$ by averaging the predictions:

$$\boldsymbol{q}_i = \frac{1}{M}\sum_{k=1}^{M}\boldsymbol{p}_i^{(k)}, \tag{3.1}$$

which represents the models' estimates of the true label. Finally, we calculate the agreement loss $\mathscr{L}_{\text{agg}}$ as the average KL divergence from $\boldsymbol{q}$ to each $\boldsymbol{p}^{(k)}, k = 1, ..., M$:

$$d(\boldsymbol{q}_i || \boldsymbol{p}_i^{(k)}) = \sum_{j=1}^{C} \boldsymbol{q}_{ij} \log \left(\frac{\boldsymbol{q}_{ij} + \varepsilon}{\boldsymbol{p}_{ij} + \varepsilon} \right), \tag{3.2}$$

$$\mathscr{L}_{\text{agg}} = \frac{1}{MN} \sum_{i=1}^{N} \sum_{k=1}^{M} d(\boldsymbol{q}_i || \boldsymbol{p}_i^{(k)}), \tag{3.3}$$

where ε is a small positive number to avoid division by zero. We can easily tell that the agreement loss encourages the models to get similar predictions based on the same input. As the KL divergence is non-negative, the agreement loss is minimized only when $\boldsymbol{q}_i = \boldsymbol{p}_i^{(k)}$ for $k = 1, ..., M$, which implies that all $\boldsymbol{p}_i^{(k)}$ should be equal because we use the average probability for $\boldsymbol{q}$. We may also use other aggregates for $\boldsymbol{q}$ as long as they satisfy that $\boldsymbol{q}_i = \boldsymbol{p}_i^{(k)}, k = 1, ..., M$ when all $\boldsymbol{p}_i^{(k)}$ are equal so as to maintain such property of the agreement loss. We consider the following alternatives for $\boldsymbol{q}$:

- **Average logits.** Given the logits $\{\boldsymbol{l}_i^{(k)}\}_{k=1}^{M}$ of predicted probabilities on the M models, we first average the logits $\boldsymbol{l}_i = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{l}_i^{(k)}$ and then feed $\boldsymbol{l}_i$ to a softmax function to get the soft target probability $\boldsymbol{q}$.

- **Max-loss probability.** A noise-robust model will disagree on noisy labels and produce large training losses. Therefore, for each instance i in the batch, we assume the prediction p_i^* that has the largest task-specific loss among the M models to be more reliable and use it as the soft target probability for instance i.

In experiments, we observe that all aggregate functions generally achieve similar performance. We present the results of different $\boldsymbol{q}$ in Section 3.4.6.

3.2.3 Joint Training

The main learning objective of our framework is then to optimize the joint loss $\mathscr{L} = \mathscr{L}_T + \gamma \mathscr{L}_{\text{agg}}$, where γ is a positive hyperparameter and $\mathscr{L}_T$ is the average of task-specific classification losses $\left\{ \mathscr{L}_{\text{sup}}^{(k)} \right\}_{k=1}^{M}$. For classification problems such as NER and RE, the task-specific loss is defined as the following cross-entropy loss, where $\boldsymbol{I}$ denotes an indicator function:

$$\mathscr{L}_{\text{sup}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \boldsymbol{I}[y_i = j] \log \boldsymbol{p}_{ij}. \tag{3.4}$$

N thereof is the number of tokens for NER and the number of sentences for RE.

The joint training can be interpreted as a "soft-pruning" scheme. For clean labels where the models' predictions are usually close to the labels, the agreement loss and its gradient are both small, so they have a small impact on training. While for noisy labels where the model predictions disagree with the training labels, the agreement loss incurs a large magnitude of gradients in training, which prevents the model from overfitting the noisy labels.

Besides co-regularization, denoising may also be attempted by "hard-pruning" the noisy labels. Small-loss selection [95, 96] assumes that instances with large task-specific loss are noisy and excludes them from training. However, some clean label instances, especially those from long-tail classes, can also have large task-specific losses and will be incorrectly pruned. While for the frequent classes, some noisy instances can have smaller task-specific losses and fail to be identified. Such errors can accumulate during training and may hinder model performance. In our framework, as we use the agreement loss instead of hard pruning, such errors will not be easily propagated (see Section 3.4.6).

3.3 Tasks

We evaluate our framework on two fundamental entity-centric IE tasks, namely RE and NER. Our framework can incorporate any kind of neural model that is dedicated to either task. Particularly, we adopt off-the-shelf SOTA models that are mainly based on Transformers. This section introduces the two attempted tasks and the design of task-specific models.

Relation extraction. RE aims at identifying the relations between a pair of entities in a piece of text from the given vocabulary of relations. Specifically, given a sentence $\boldsymbol{x}$ and two entities e_s and e_o, identified as the subject and object entities respectively, the goal is to predict the relation between e_s and e_o. Following Shi & Lin [16], we formulate this task as a sentence classification problem. Accordingly, we first apply the entity masking technique [20] to the input sentence and replace the subject and object entities with their named entity types. For example, a short sentence "*Bill Gates founded Microsoft*" will become "[SUBJECT-PERSON] *founded* [OBJECT-ORGANIZATION]" after entity masking. We then feed the sentence to the pre-trained language model and use a softmax classifier on the representation of the [CLS] token to predict the relation.

Named entity recognition. NER seeks to locate and classify named entities in text into pre-defined categories. Following Devlin, Chang, Lee & Toutanova [14], we formulate the task as a token classification problem. In detail, a Transformer-based language model first tokenizes an input sentence into a sub-token sequence. To classify each token, the representation of its first sub-token is sent into a softmax classifier. We use the BIO tagging scheme [97] and output the tag with the maximum likelihood as the predicted label.

3.4 Experiment

In this section, we evaluate the proposed learning framework based on two (noisy) benchmark datasets for the two entity-centric IE tasks (Section 3.4.1-Section 3.4.4). In addition, a noise filtering analysis is presented to show how our framework prevents an incorporated neural model from overfitting noisy training data (Section 3.4.5), along with a detailed ablation study about configurations with varied model copies, alternative noise filtering strategies, target functions, and different noise rates (Section 3.4.6).

Dataset	# train	# dev	# test	# classes	% noise
TACRED	68124	22631	15509	42	6.62
CoNLL03	14041	3250	3453	9	5.38

Table 3.1: Data statistics of TACRED and CoNLL03.

3.4.1 Datasets

The experiments are conducted on TACRED [20] and CoNLL03 [85]. TACRED is a crowdsourced dataset for relation extraction. A recent study by Alt, Gabryszak & Hennig [87] found a large portion of examples to be mislabeled and rectified some incorrect labels in the development and test sets. CoNLL03 is a human-annotated dataset for NER. Another study by Wang, Shang, Liu, Lu, Liu & Han [93] found that in 5.38% of sentences in CoNLL03, at least one token is mislabeled. Accordingly, Wang, Shang, Liu, Lu, Liu & Han [93] also relabeled the test set[2]. We summarize the statistics of both datasets in Table 3.1. For all compared methods, we report the results on both the original and relabeled evaluation sets.

3.4.2 Base Models

We evaluate our framework by incorporating the following SOTA models:

- **C-GCN** [30] is a graph-based model for RE. It prunes the dependency graph and applies graph convolutional networks to get the representation of entities.

- **BERT** [14] is a Transformer-based language model that is pre-trained from large-scale text corpora. Both `Base` and `Large` versions of the model are considered in our experiments.

- **LUKE** [98] is a Transformer-based language model that is pre-trained on both large-scale text corpora and knowledge graphs. It achieves SOTA performance on various entity-related tasks, including RE and NER.

We report the performance of the base models trained with and without our co-regularization framework. We also compare our framework to CrossWeigh [93], which is another noisy-label learning framework. Specifically, CrossWeigh partitions the training set into equal-sized chunks, reserves each chunk, and then trains several models on the rest ones. After training, the models predict on the reserved chunk, and instances on which the models disagree are down-weighted. In the end, the chunks are combined and used to train a new model for inference. Learning by CrossWeigh is dependant on a high computation cost. Wang, Shang, Liu, Lu, Liu & Han [93] split the CoNLL03 dataset into 10 chunks and train 3 models on each partition, resulting in a total of number 30 models. We follow their settings and train 30 models on both TACRED and CoNLL03.

[2]Note that neither of these two datasets comes with a relabeled (clean) training set. All models are still trained on the original noisy training set.

Model	Original		Relabeled	
	Dev F_1	Test F_1	Dev F_1	Test F_1
C-GCN ♣ [30]	67.2	66.7	74.9	74.6
C-GCN-CrossWeigh	67.8	67.4	75.6	75.7
C-GCN-CR	67.7	67.2	75.6	75.4
$\text{BERT}_{\text{BASE}}$ [14]	69.1	68.9	76.4	76.9
$\text{BERT}_{\text{BASE}}$-CrossWeigh	71.3	70.8	79.2	79.1
$\text{BERT}_{\text{BASE}}$-CR	71.5	71.1	79.9	80.0
$\text{BERT}_{\text{LARGE}}$ [14]	70.9	70.2	78.3	77.9
$\text{BERT}_{\text{LARGE}}$-CrossWeigh	72.1	71.9	79.5	79.8
$\text{BERT}_{\text{LARGE}}$-CR	73.1	73.0	81.3	82.0
LUKE ♣ [98]	71.1	70.9	80.1	80.6
LUKE-CrossWeigh	71.0	71.6	80.4	81.6
LUKE-CR	71.8	72.4	81.9	83.1

Table 3.2: F_1 score (%) on the dev and test set of TACRED. ♣ marks results obtained from the originally released implementation. We report the median of F_1 on 5 runs of training using different random seeds. For fair comparison, the CR results are reported based on the predictions from model f_1 in our framework.

3.4.3 Model Configurations

For base models C-GCN [30] and LUKE [98], we rerun the officially released implementations using the recommended hyperparameters in the original papers. We implement $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ based on Huggingface's Transformers [99]. For CrossWeigh [93], we re-implement this framework using those compared base models. All models are optimized with Adam [100] using a learning rate of 6e$-$5 for TACRED and that of 1e$-$5 for CoNLL03, with a linear learning rate decay to 0. The batch size is fixed as 64 for all models. We finetune the TACRED model for 5 epochs and the CoNLL03 model for 50 epochs. The best model checkpoint is chosen based on the F_1 score on the development set. We tune γ from $\{1.0, 2.0, 5.0, 10.0, 20.0\}$, and tune α from $\{10, 30, 50, 70, 90\}$. We report the median of F_1 of 5 runs using different random seeds.

For efficiency, we use the simplest setup of our framework with two model copies ($M = 2$) in the main experiments (Section 3.4.4). $\boldsymbol{q}$ is set as the average probability in the main experiment. Performance with more model copies and alternative aggregates is later studied in Section 3.4.6.

3.4.4 Main Results

The experiment results on TACRED and CoNLL03 are reported in Table 3.2 and Table 3.3 respectively, where methods incorporated in our learning framework are marked with "CR". As stated, the results are reported under the setup where $M = 2$. For a fair comparison, the results are reported based on the predictions from model f_1 in the framework. On TACRED, our framework leads to an absolute improvement of $2.5 - 4.1\%$ in F_1 on the relabeled test set for Transformer-based models, and a relatively smaller gain (0.8% in F_1) for C-GCN. In particular, our framework enhances the

Model	Original		Relabeled
	Dev F_1	Test F_1	Test F_1
$\text{BERT}_{\text{BASE}}$ [14]	95.58	91.96	92.91
$\text{BERT}_{\text{BASE}}$-CrossWeigh	95.65	92.15	93.03
$\text{BERT}_{\text{BASE}}$-CR	95.87	92.53	93.48
$\text{BERT}_{\text{LARGE}}$ [14]	96.16	92.24	93.22
$\text{BERT}_{\text{LARGE}}$-CrossWeigh	96.32	92.49	93.61
$\text{BERT}_{\text{LARGE}}$-CR	96.59	92.82	94.04
LUKE ♣ [98]	97.03	93.91	95.60
LUKE–CrossWeigh	97.09	93.98	95.75
LUKE-CR	97.21	94.22	95.88

Table 3.3: F_1 score (%) on the dev and test set of CoNLL03. ♣ marks results obtained using the originally released code.

SOTA method LUKE by 2.5% in F_1, leading to a very promising F_1 score of 83.1%. On CoNLL03, where the noise rate is smaller than TACRED, our framework leads to a performance gain of $0.28-0.82\%$ in F_1 on the relabeled test set. On both IE tasks, our framework also leads to a consistent improvement on the original test set. Compared to CrossWeigh, except for C-GCN where the results are similar, our framework consistently outperforms it by $0.9-2.2\%$ on TACRED and by $0.13-0.45\%$ on CoNLL03. Moreover, as our framework requires training M models concurrently while CrossWeigh requires training redundant models (30 in experiments), the computation cost of our co-regularization framework is much lower than CrossWeigh. In general, the results here show the effectiveness and practicality of the proposed framework.

3.4.5 Noise Filtering Analysis

The main experiments show that our framework can improve the overall performance of models trained with noisy labels. In this section, we further demonstrate how our framework prevents overfitting on noisy labels. To do so, we extract the 2,526 noisy instances from the development and test sets of TACRED where the relabeling by Alt, Gabryszak & Hennig [87] disagrees with the original labels. Accordingly, we obtain a *noisy set* containing those examples with original labels and a *clean set* with rectified labels. We train a relation classifier on the union of the training set and the noisy set and then evaluate the model on the clean set. In this case, worse performance on the clean set indicates more severe overfitting on noisy labels.

Figure 3.2 shows the results by C-GCN-CR and $\text{BERT}_{\text{BASE}}$-CR on the clean set, where we observe that: (1) Compared to the original base models ($\gamma = 0.0$), those trained with our framework achieves higher F_1 scores, indicating improved robustness against the label noise; (2) Comparing different base models, the large classifier $\text{BERT}_{\text{BASE}}$ is typically less noise-robust than a smaller model like C-GCN, which explains why the performance gain from our framework is more notable on $\text{BERT}_{\text{BASE}}$; (3) For both models, the F_1 score first increases then decreases, consistent with the delayed learning curves that the neural models have on noisy instances [81].

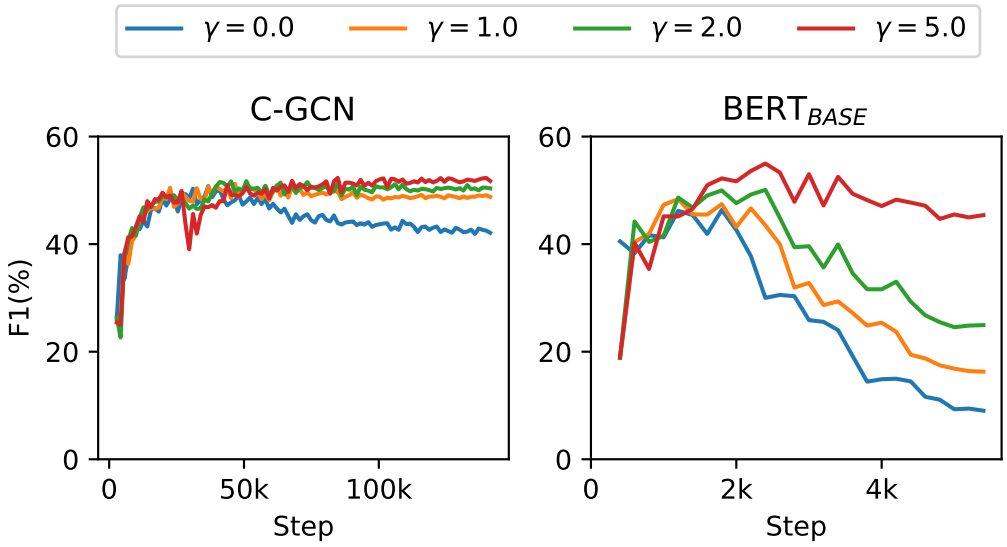

Figure 3.2: F_1 score (%) on the clean set of TACRED. Classifiers trained with our framework are more noise-robust compared to baselines ($\gamma = 0$).

# Models	2	3	4
BERT$_{\text{BASE}}$-CR	80.0	79.5	79.8
BERT$_{\text{LARGE}}$-CR	82.0	82.4	82.7

Table 3.4: F_1 score (%) of using different number of models on the relabeled test set of TACRED.

3.4.6 Ablation Study

Using extra model copies. The main results show that using two copies of a model in the co-regularization framework has already improved the performance by a remarkable margin. Intuitively, more models may generate higher-quality soft target probabilities and thus further improve the performance. We further show the performance on TACRED by incorporating more model copies. We report the relabeled test F_1 on TACRED in Table 3.4. We observe that increasing the number of copies does not necessarily lead to a notable increase in performance. On BERT$_{\text{LARGE}}$, increasing the number of model copies from 2 to 4 gradually improves the performance from 82.0% to 82.7%. While on BERT$_{\text{BASE}}$, increasing the number of model copies does not improve the performance. We notice that the increased number of copies leads to a significant increase in the agreement loss for BERT$_{\text{BASE}}$, indicating that the copies of BERT$_{\text{BASE}}$ fail to reach a consensus based on the same input. This may be due to the relatively small model capacity of BERT$_{\text{BASE}}$. Overall, this study shows that the optimal M is dependent on the models and needs to be tuned on the specific task. Note that as the models can be trained in parallel, increasing the number of models does not necessarily increase the training time, though being at the cost of more computational resources.

Alternative strategies for noise filtering. Besides co-regularization, we also experiment with other noise-filtering strategies. Small-loss selection [95, 96, 101] prunes the instances with the largest training losses in the training batches. This method is motivated by the fact that the noisy instances take a longer time to be memorized and usually cause a large training loss. We further try another

Model	Original Test F_1	Relabeled Test F_1
$\text{BERT}_{\text{BASE}}$	68.9	76.9
$\text{BERT}_{\text{BASE}}$ + Small-loss selection	68.7	76.6
$\text{BERT}_{\text{BASE}}$ + Relabeling	69.0	77.7

Table 3.5: F_1 score (%) of alternative noise filtering strategies on the test set of TACRED. The best results are achieved when $\delta = 2\%$ for both methods.

Functions	Avg prob	Avg logit	Max-loss prob
$\text{BERT}_{\text{BASE}}$-CR	80.0	79.9	79.4
$\text{BERT}_{\text{LARGE}}$-CR	82.0	81.6	82.2

Table 3.6: F_1 score (%) of different functions for $\boldsymbol{q}$ on the relabeled test set of TACRED.

strategy named relabeling. Instead of pruning the large-loss training instances, we relabel them with the most likely labels from model predictions.

We evaluate the two noise filtering strategies on TACRED using $\text{BERT}_{\text{BASE}}$ as the base model. For both strategies, we prune/relabel $\delta_t = \delta \cdot \frac{t}{T}$ percent of examples with the largest training loss in each training batch following Han, Yao, Yu, Niu, Xu, Hu, *et al.* [96], where t is the current number of training steps, T is the total number of training steps, and δ is the maximum pruning/relabeling rate. These hyperparameters are tuned on the development set. The training loss is defined as the average task-specific loss of the M models, where we set $M = 2$ in consistent with the main experiments (Section 3.4.4). We try δ from $\{2\%, 5\%, 8\%\}$ and report the best results.

Results are shown in Table 3.5. We find that $\delta = 2\%$ achieves the best performance for both strategies. The small-loss selection strategy underperforms the base model without noise filtering. Relabeling outperforms the base model slightly, but the improvements are lesser than the proposed co-regularization method. We observe that these two strategies do not work well on imbalanced datasets, mostly pruning or relabeling training examples from long-tail classes. Specifically, on the TACRED dataset, where the NA class accounts for 80% of the total labels, only 20% pruned labels are from NA while the remaining 80% are from other classes. It is because that the model's predictions will be biased towards the frequent classes on imbalanced datasets, therefore leading to the large training loss on long-tail instances. Once pruned or relabeled, such long-tail instances are excluded from training, causing further error propagation that can lead to more biased predictions. Our framework, on the contrary, adopts an agreement loss instead of hard pruning or relabeling, which reduces such error propagation.

Alternative aggregates for $\boldsymbol{q}$. Besides the average probability, we evaluate two other aggregates for $\boldsymbol{q}$, i.e. the average logits and the max-loss probability (Section 3.2.2). This experiment is conducted with $M = 2$. F_1 results on the relabeled TACRED test set (Table 3.6) suggest that different aggregates generally achieve comparable performance,with a marginal difference of up to 0.6% in F_1. Therefore, the default setup is suggested to be the average probability, which is easier to implement.

Flipped labels (%)	10	30	50	70	90
$\text{BERT}_{\text{BASE}}$	74.2	70.8	62.9	48.6	0
$\text{BERT}_{\text{BASE}}$-CrossWeigh	77.3	75.6	71.6	61.3	25.1
$\text{BERT}_{\text{BASE}}$-CR	79.3	78.3	73.2	63.5	34.1
$\text{BERT}_{\text{BASE}}$ w/o flipped labels	76.5	74.9	72.9	70.8	57.4

Table 3.7: F_1 score (%) under different noise rates on the relabeled set of TACRED.

Performance under different noise rates. We further evaluate our framework on training data of different noise rates. To do so, we create noisy training data by randomly flipping 10%, 30%, 50%, 70%, or 90% labels in the training set of TACRED. Then we use those synthetic noisy training sets to train RE models and evaluate them on the relabeled test set of TACRED. We use $\text{BERT}_{\text{BASE}}$ as the base model and report the median F_1 score of 5 trials. Results are given in Table 3.7, which show that our co-regularization framework consistently outperforms both the base model and CrossWeigh under different noise rates. The gain generally becomes larger as the noise rate increases. In comparison to $\text{BERT}_{\text{BASE}}$ trained on the training sets where all flipped labels are removed, our framework, even trained on synthetic noise, achieves comparable or better results when the noise rates are below 50%.

3.5 Related Work

We discuss two lines of related work. Each has a large body of work which we can only provide as a highly selected summary.

Distant supervision. Distant supervision [22, 102] generates noisy training data with heuristics to align unlabeled data with labels, whereas much effort has been devoted to reducing labeling noise. Multi-instance learning [23, 103–105] creates bags of noisily labeled instances and assumes at least one instance in each bag is correct, then it uses heuristics or auxiliary classifiers to select the correct labels. However, such instance bags may not exist in a general supervised setting. Reinforcement learning [92, 106, 107] and curricular learning [90, 95] methods use a clean validation set to obtain an auxiliary model for noise filtering, while constructing a perfectly labeled validation set is expensive. Our framework can learn noise-robust IE models without extra learning resources and can be easily incorporated into existing supervised IE models.

Supervised learning with noisy labels. A deep neural network can memorize noisy labels, and its generalizability will severely degrade when trained with noisy labels [82]. In computer vision, much investigation has been conducted for supervised image classification with noise, producing techniques such as robust loss functions [108–111], noise filtering layers [112, 113], label re-weighting [114, 115], robust regularization [63, 116–119], and sample selection [95, 96, 120–122]. The robust loss functions and noise filtering layers require modifying model structures and may not be easily adapted to IE models. The sample selection methods assume the data instances with large training losses to be noisy and exclude them from training. However, some clean instances, especially those from long-tail classes, can also have a large training loss and be wrongly pruned, leading to propagated errors.

In NLP, few efforts have focused on learning with denoising. CrossWeigh [93], one label re-weighting method, partitions the training data into multiple folds and trains multiple models on each fold. Instances on which models disagree are regarded as noisy and down-weighted in training. However, this method requires training many models and is computationally expensive. Our framework only requires training several models concurrently, which is more computationally efficient and achieves better performance. NetAb [123] assumes that noisy labels are created from randomly flipping clean labels and uses a CNN to model the noise transition matrix [124]. However, this assumption does not hold for real datasets, where the noise rate vary among data instances [125].

3.6 Conclusion

This chapter presents a co-regularization framework for learning supervised IE models from noisy data. This framework consists of two or more identically structured models with different initialization, which are encouraged to give similar predictions on the same inputs by optimizing an agreement loss. On noisy examples where model predictions usually differ from the labels, the agreement loss prevents the model from overfitting noisy labels. Experiments on NER and RE benchmarks show that our framework yields promising improvements on various IE models. For future work, we plan to extend the use of the proposed framework to other tasks such as event-centric IE [126] and co-reference resolution [127].

Chapter 4

Out-of-distribution Detection

Pretrained Transformers achieve remarkable performance when training and test data are from the same distribution. However, in real-world scenarios, the model often faces out-of-distribution (OOD) instances that can cause severe semantic shift problems at inference time. Therefore, in practice, a reliable model should identify such instances, and then either reject them during inference or pass them over to models that handle another distribution. We develop an unsupervised OOD detection method, in which only the in-distribution (ID) data are used in training. We propose to fine-tune the Transformers with a contrastive loss, which improves the compactness of representations, such that OOD instances can be better differentiated from ID ones. These OOD instances can then be accurately detected using the Mahalanobis distance in the model's penultimate layer. We experiment with comprehensive settings and achieve near-perfect OOD detection performance, outperforming baselines drastically. We further investigate the rationales behind the improvement, finding that more compact representations through margin-based contrastive learning bring improvement[1].

4.1 Introduction

Many natural language classifiers are developed based on a closed-world assumption, i.e., the training and test data are sampled from the same distribution. However, training data can rarely capture the entire distribution. In real-world scenarios, out-of-distribution (OOD) instances, which come from categories that are not known to the model, can often be present in inference phases. These instances could be misclassified by the model into known categories with high confidence, causing the semantic shift problem [129]. As a practical solution to this problem in real-world applications, the model should detect such instances, and signal exceptions or transmit to models handling other categories or tasks. Although pretrained Transformers [14] achieve remarkable results when intrinsically evaluated on in-distribution (ID) data, recent work [1] shows that many of these models fall short of detecting OOD instances.

Despite the importance, few attempts have been made for the problem of detecting OOD in NLP tasks. One proposed method is to train a model on both the ID and OOD data and regularize the model to produce lower confidence on OOD instances than ID ones [130, 131]. However, as the OOD instances reside in an unbounded feature space, their distribution during inference is usually unknown. Hence, it is hard to decide which OOD instances to use in training, let alone that they may not be available in lots of scenarios. Another practiced method for OOD detection is to use the

[1]This chapter is based on Zhou, Liu & Chen [128].

maximum class probability as an indicator [1, 132], such that lower values indicate more probable OOD instances. Though easy to implement, its OOD detection performance is far from perfection, as prior studies [133, 134] show that OOD inputs can often get high probabilities as well.

We aim at improving the OOD detection ability of natural language classifiers, in particular, the pretrained Transformers, which have been the backbones of many SOTA NLP systems. For practical purposes, we adopt the setting where only ID data are available during task-specific training. Moreover, we require that the model should maintain classification performance on the ID task data. To this end, we propose a contrastive learning framework for unsupervised OOD detection, which is composed of a contrastive loss and an OOD scoring function. Our contrastive loss aims at increasing the discrepancy of the representations of instances from different classes in the task. During training, instances belonging to the same class are regarded as pseudo-ID data while those of different classes are considered mutually pseudo-OOD data. We hypothesize that increasing inter-class discrepancies can help the model learn discriminative features for ID/OOD distinctions, and therefore help detect true OOD data at inference. We study two versions of the contrastive loss: a similarity-based contrastive loss [135–137] and a margin-based contrastive loss. The OOD scoring function maps the representations of instances to OOD detection scores, indicating the likelihood of an instance being OOD. We examine different combinations of contrastive losses and OOD scoring functions, including maximum softmax probability, energy score, Mahalanobis distance, and maximum cosine similarity. Particularly, we observe that OOD scoring based on the Mahalanobis distance [138], when incorporated with the margin-based contrastive loss, generally leads to the best OOD detection performance. The Mahalanobis distance is computed from the penultimate layer[2] of Transformers by fitting a class-conditional multivariate Gaussian distribution.

The main contributions of this work are three-fold. First, we propose a contrastive learning framework for unsupervised OOD detection, where we comprehensively study combinations of different contrastive learning losses and OOD scoring functions. Second, extensive experiments on various tasks and datasets demonstrate the significant improvement our method has made to OOD detection for Transformers. Third, we provide a detailed analysis to reveal the importance of different incorporated techniques, which also identifies further challenges for this emerging research topic.

4.2 Related Work

Out-of-Distribution Detection. Determining whether an instance is OOD is critical for the safe deployment of machine learning systems in the real world [139]. The main challenge is that the distribution of OOD data is hard to estimate *a priori*. Based on the availability of OOD data, recent methods can be categorized into supervised, self-supervised, and unsupervised ones. Supervised methods train models on both ID and OOD data, where the models are expected to output a uniform distribution over known classes on OOD data [130, 133, 140]. However, it is hard to assume the presence of a large dataset that provides comprehensive coverage for OOD instances in practice. Self-supervised methods [141] apply augmentation techniques to change certain properties of data (e.g., through rotation of an image) and simultaneously learn an auxiliary model to predict the property changes (e.g., the rotation angle). Such an auxiliary model is expected to have worse

[2] I.e., the input to the softmax layer.

generalization on OOD data which can in turn be identified by a larger loss. However, it is hard to define such transformations for natural language. Unsupervised methods use only ID data in training. They detect OOD data based on the class probabilities [132, 134, 142, 143] or other latent space metrics [138, 144]. Particularly, Vyas, Jammalamadaka, Zhu, Das, Kaul & Willke [145] randomly split the training classes into two subsets and treat them as pseudo-ID and pseudo-OOD data, respectively. They then train an OOD detector that requires the entropy of probability distribution on pseudo-OOD data to be lower than pseudo-ID data. This process is repeated to obtain multiple OOD detectors, and their ensemble is used to detect the OOD instances. This method conducts OOD detection at the cost of high computational overhead in training redundant models and has the limitation of not supporting the detection for binary classification tasks.

Though extensively studied for computer vision (CV), OOD detection has been overlooked in NLP, and most prior works [130, 146, 147] require both ID and OOD data in training. Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1] use the maximum softmax probability as the detection score and show that pretrained Transformers exhibit better OOD detection performance than models such as LSTM [38], while the performance is still imperfect. Our framework, as an unsupervised OOD detection approach, significantly improves the OOD detection of Transformers only using ID data.

Contrastive Learning. Recently, contrastive learning has received a lot of research attention. It works by mapping instances of the same class into a nearby region and make instances of different classes uniformly distributed [148]. Many efforts on CV [137, 149, 150] and NLP [151] incorporate contrastive learning into self-supervised learning, which seeks to gather the representations of different augmented views of the same instance and separate those of different instances. Prior work on image classification [152, 153] shows that model trained with self-supervised contrastive learning generates discriminative features for detecting distributional shifts. However, such methods heavily rely on data augmentation of instances and are hard to be applied to NLP. Other efforts on CV [154] and NLP [155] conduct contrastive learning in a supervised manner, which aims at embedding instances of the same class closer and separating different classes. They show that models trained with supervised contrastive learning exhibit better classification performance. To the best of our knowledge, we are the first to introduce supervised contrastive learning to OOD detection. Such a method does not rely on data augmentation, thus can be easily adapted to existing NLP models. We also propose a margin-based contrastive objective that greatly outperforms standard supervised contrastive losses.

4.3 Method

In this section, we first formally define the OOD detection problem (Section 4.3.1), then introduce the overall framework (Section 4.3.2), and finally present the contrastive representation learning and scoring functions (Section 4.3.3 and Section 4.3.4).

4.3.1 Problem Definition

We aim at improving the OOD detection performance of natural language classifiers that are based on pretrained Transformers, using only ID data in the main-task training. Generally, the out-of-distribution (OOD) instances can be defined as instances $(\boldsymbol{x}, y)$ sampled from an underlying

distribution other than the training distribution $\mathrm{P}(\mathscr{X}_{\text{train}}, \mathscr{Y}_{\text{train}})$, where $\mathscr{X}_{\text{train}}$ and $\mathscr{Y}_{\text{train}}$ are the training corpus and training label set, respectively. In this context, literature further divides OOD data into those with *semantic shift* or *non-semantic shift* [129]. *Semantic shift* refers to the instances that do not belong to $\mathscr{Y}_{\text{train}}$. More specifically, instances with semantic shift may come from unknown categories or irrelevant tasks. Therefore, the model is expected to detect and reject such instances (or forward them to models handling other tasks), instead of mistakenly classifying them into $\mathscr{Y}_{\text{train}}$. *Non-semantic shift*, on the other hand, refers to the instances that belong to $\mathscr{Y}_{\text{train}}$ but are sampled from a distribution other than $\mathscr{X}_{\text{train}}$, e.g., a different corpus. Though drawn from OOD, those instances can be classified into $\mathscr{Y}_{\text{train}}$, thus can be accepted by the model. Hence, in the context of this thesis, we primarily consider an instance $(\boldsymbol{x}, y)$ to be OOD if $y \notin \mathscr{Y}_{\text{train}}$, i.e., exhibiting semantic shift, to be consistent with the problem settings of prior studies [1, 138, 143].

We hereby formally define the OOD detection task. Specifically, given a main task of natural language classification (e.g., sentence classification, NLI, etc.), for an instance $\boldsymbol{x}$ to be classified, our goal is to develop an auxiliary OOD scoring function $f(\boldsymbol{x}) : \mathscr{X} \to \mathbb{R}$. This function should return a low score for an ID instance where $y \in \mathscr{Y}_{\text{train}}$, and a high score for an OOD instance where $y \notin \mathscr{Y}_{\text{train}}$ (y is the underlying label for $\boldsymbol{x}$ and is unknown at inference). During inference, we can set a threshold for the OOD score to filter out most OOD instances. This process involves a trade-off between false negative and false positive and may be specific to the application. Meanwhile, we expect that the OOD detection auxiliary should not negatively affect the performance of the main task on ID data.

4.3.2 Framework Overview

Next, we introduce the formation of our contrastive learning framework for OOD detection. We decompose OOD detection into two steps. The first step is contrastive representation learning, where we focus on learning a representation space $\mathscr{H}$ where the distribution of ID and that of OOD data are distinct. Accordingly, we need another function to map the representation to an OOD score. This process is equivalent to expressing OOD detection as $f(\boldsymbol{x}) = g(\boldsymbol{h})$, where $\boldsymbol{h} \in \mathscr{H}$ is the dense representation of the input text $\boldsymbol{x}$ given by an encoder, $g : \mathscr{H} \to \mathbb{R}$ is a scoring function mapping the representation to an OOD detection score. Using this decomposition, we can use different training strategies for $\boldsymbol{h}$ and different functions for g, which are studies in the following sections.

The learning process of our framework is described in Algorithm 2. In the training phase, our framework takes training and validation datasets that are both ID as input. The model is optimized with both the (main task) classification loss and the contrastive loss on batches sampled from ID training data. The best model is selected based on the ID validation data. Specifically, for a distribution-based OOD scoring function such as the Mahalanobis distance, we first need to fit the OOD detector on the ID validation data. We then evaluate the trained model on the ID validation data, where a satisfactory model should have a low contrastive loss and preserve the classification performance. In the end, our framework returns a classifier to handle the main task on ID data and an OOD detector to identify OOD instances at inference.

4.3.3 Contrastive Representation Learning

In this section, we discuss how to learn distinctive representations for OOD detection. For better OOD detection performance, the representation space $\mathscr{H}$ is supposed to minimize the overlap of

Algorithm 2: Learning Process

Input: ID training set $\mathscr{D}_{\text{train}}$ and ID validation set $\mathscr{D}_{\text{val}}$.
Output: A trained classifier and an OOD detector.
Initialize the pretrained Transformer M.
for $t = 1...T$ **do**
  Sample a batch from $\mathscr{D}_{\text{train}}$.
  Calculate the classification loss $\mathscr{L}_{\text{ce}}$.
  Calculate the contrastive loss $\mathscr{L}_{\text{cont}}$ as either $\mathscr{L}_{\text{scl}}$ or $\mathscr{L}_{\text{margin}}$
  $\mathscr{L} = \mathscr{L}_{\text{ce}} + \lambda \mathscr{L}_{\text{cont}}$.
  Update model parameters w.r.t. $\mathscr{L}$.
  if t % *evaluation steps* $= 0$ **then**
    Fit the OOD detector on $\mathscr{D}_{\text{val}}$.
    Evaluate both the classifier and OOD detector on $\mathscr{D}_{\text{val}}$.
Return the best model checkpoint.

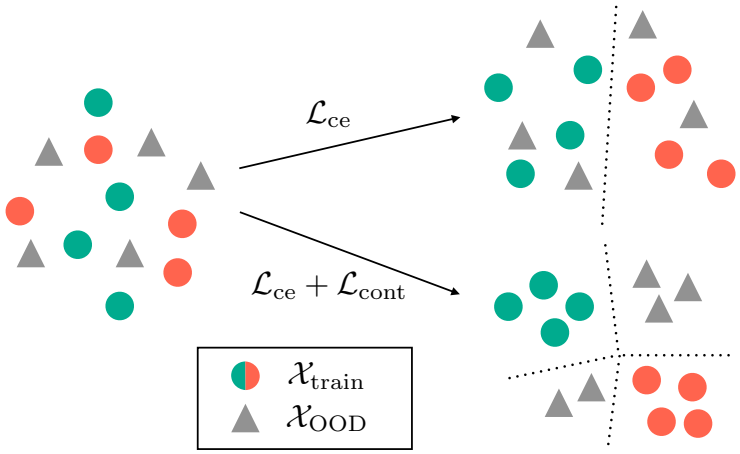

Figure 4.1: Illustration of our proposed contrastive loss. The contrastive loss seeks to increase the discrepancy of the representations for instances from different training classes, such that OOD instances from unknown classes can be better differentiated.

the representations of ID and OOD data. In a supervised setting where both ID and OOD data are available in training, it would be easy to obtain such $\mathscr{H}$. For example, Dhamija, Günther & Boult [133] train the neural model on both ID and OOD data and require the magnitude of representations of OOD instances to be smaller than ID representations. However, in real-world applications, the distribution of OOD data is usually unknown beforehand. We thus tackle a more general problem setting where the OOD data are assumed unavailable in training (unsupervised OOD detection, introduced below).

In this unsupervised setup, though all training data used are ID, they may belong to different classes. We leverage data of distinct classes to learn more discriminative features. Through a contrastive learning objective, instances of the same class form compact clusters, while instances of different classes are encouraged to live apart from each other beyond a certain margin, as illustrated in Figure 4.1. The discriminative feature space is generalizable to OOD data, which ultimately leads to better OOD detection performance in inference when encountering an unknown distribution. We

realize such a strategy using two alternatives of contrastive losses, i.e., the *supervised contrastive loss* and the *margin-based contrastive loss*.

Supervised Contrastive Loss. Different from the contrastive loss used in self-supervised representation learning [137, 150] that compares augmented instances to other instances, our contrastive loss contrasts instances to those from different ID classes. To give a more specific illustration of our technique, we first consider the supervised contrastive loss [154, 155]. Specifically, for a multi-class classification problem with C classes, given a batch of training instances $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$, where $\boldsymbol{x}_i$ is the input text, y_i is the ground-truth label, the supervised contrastive loss can be formulated as:

$$\mathscr{L}_{\text{scl}} = \sum_{i=1}^{M} \frac{-1}{M|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\boldsymbol{z}_i^\top \boldsymbol{z}_p / \tau}}{\sum\limits_{a \in A(i)} e^{\boldsymbol{z}_i^\top \boldsymbol{z}_a / \tau}},$$

where $A(i) = \{1, ..., M\} \backslash \{i\}$ is the set of all anchor instances, $P(i) = \{p \in A(i) : y_i = y_p\}$ is the set of anchor instances from the same class as i, τ is a temperature hyper-parameter, $\boldsymbol{z}$ is the L2-normalized `[CLS]` embedding before the softmax layer [154, 155]. The L2 normalization is for avoiding huge values in the dot product, which may lead to unstable updates. In this case, this loss is optimized to increase the cosine similarity of instance pairs if they are from the same class and decrease it otherwise.

Margin-based Contrastive Loss. The supervised contrastive loss produces minimal gradients when the similarity difference of positive and negative instances exceeds a certain point. However, to better separate OOD instances, it is beneficial to enlarge the discrepancy between classes as much as possible. Therefore, we propose another margin-based contrastive loss. It encourages the L2 distances of instances from the same class to be as small as possible, forming compact clusters, and the L2 distances of instances from different classes to be larger than a margin. Our loss is formulated as:

$$\begin{aligned}
\mathscr{L}_{\text{pos}} &= \sum_{i=1}^{M} \frac{1}{|P(i)|} \sum_{p \in P(i)} \|\boldsymbol{h}_i - \boldsymbol{h}_p\|^2, \\
\mathscr{L}_{\text{neg}} &= \sum_{i=1}^{M} \frac{1}{|N(i)|} \sum_{n \in N(i)} (\xi - \|\boldsymbol{h}_i - \boldsymbol{h}_n\|^2)_+, \\
\mathscr{L}_{\text{margin}} &= \frac{1}{dM} \left(\mathscr{L}_{\text{pos}} + \mathscr{L}_{\text{neg}} \right).
\end{aligned}$$

Here $N(i) = \{n \in A(i) : y_i \neq y_n\}$ is the set of anchor instances from other classes than y_i, $\boldsymbol{h} \in \mathbb{R}^d$ is the unnormalized `[CLS]` embedding before the softmax layer, ξ is a margin, d is the number of dimensions of $\boldsymbol{h}$. As we do not use OOD data in training, it is hard to properly tune the margin. Hence, we further incorporate an adaptive margin. Intuitively, distances between instances from the same class should be smaller than those from different classes. Therefore, we define the margin as the maximum distance between pairs of instances from the same class in the batch:

$$\xi = \max_{i=1}^{M} \max_{p \in P(i)} \|\boldsymbol{h}_i - \boldsymbol{h}_p\|^2.$$

We evaluate both contrastive losses in experiments. In training, the model is jointly optimized with the cross-entropy classification loss $\mathscr{L}_{\text{ce}}$ and the contrastive loss $\mathscr{L}_{\text{cont}}$:

$$\mathscr{L} = \mathscr{L}_{\text{ce}} + \lambda \mathscr{L}_{\text{cont}},$$

where λ is a positive coefficient. We tune λ based on the contrastive loss and the classification performance on the ID validation set, where a selected value for λ should achieve a smaller contrastive loss while maintaining the classification performance.

4.3.4 OOD Scoring Functions

Next, we introduce the modeling of the OOD scoring function g. The goal of the scoring function g is to map the representations of instances to OOD detection scores, where higher scores indicate higher likelihoods for being OOD. In the following, we describe several choices of this scoring function.

Maximum Softmax Probability (MSP). Hendrycks & Gimpel [143] use the maximum class probability $1 - \max_{j=1}^{C} \boldsymbol{p}_j$ among C training classes in the softmax layer as an OOD indicator. This method has been widely adopted as a baseline for OOD detection [1, 129, 141, 143].

Energy Score (Energy). Liu, Wang, Owens & Li [144] interpret the softmax function as the ratio of the joint probability in $\mathscr{X} \times \mathscr{Y}$ to the probability in $\mathscr{X}$, and estimates the probability density of inputs as:

$$g = -\log \sum_{j=1}^{C} \exp(\boldsymbol{w}_j^{\intercal} \boldsymbol{h}),$$

where $\boldsymbol{w}_j \in \mathbb{R}^d$ is the weight of the j^{th} class in the softmax layer, $\boldsymbol{h}$ is the input to the softmax layer. A higher g means lower probability density in ID classes and thus implies higher OOD likelihood.

Mahalanobis Distance (Maha). Lee, Lee, Lee & Shin [138] model the ID features with class-conditional multivariate Gaussian distributions. It first fits the Gaussian distributions on the ID validation set $\mathscr{D}_{\text{val}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ using the input representation $\boldsymbol{h}$ in the penultimate layer of model:

$$\begin{aligned} \boldsymbol{\mu}_j &= \mathbb{E}_{y_i=j}\left[\boldsymbol{h}_i\right], j = 1, ..., C, \\ \boldsymbol{\Sigma} &= \mathbb{E}\left[\left(\boldsymbol{h}_i - \boldsymbol{\mu}_{y_i}\right)\left(\boldsymbol{h}_i - \boldsymbol{\mu}_{y_i}\right)^{\intercal}\right], \end{aligned}$$

where C is the number of classes, $\boldsymbol{\mu}_j$ is the mean vector of classes, and $\boldsymbol{\Sigma}$ is a shared covariance matrix of all classes. Then, given an instance $\boldsymbol{x}$ during inference, it calculates the OOD detection score as the minimum Mahalanobis distance among the C classes:

$$g = -\min_{j=1}^{C} (\boldsymbol{h} - \boldsymbol{\mu}_j)^{\intercal} \boldsymbol{\Sigma}^{+} (\boldsymbol{h} - \boldsymbol{\mu}_j),$$

where $\boldsymbol{\Sigma}^{+}$ is the pseudo-inverse of $\boldsymbol{\Sigma}$. The Mahalanobis distance calculates the probability density of $\boldsymbol{h}$ in the Gaussian distribution.

Cosine Similarity can also be incorporated to consider the angular similarity of input representations. To do so, the scoring function returns the OOD score as the maximum cosine similarity of $\boldsymbol{h}$ to instances of the ID validation set:

$$g = -\max_{i=1}^{M} \cos(\boldsymbol{h}, \boldsymbol{h}_i^{(val)}).$$

The above OOD scoring functions, combined with options of contrastive losses, lead to different variants of our framework. We evaluate each combination in experiments.

4.4 Experiments

This section presents experimental evaluations of the proposed OOD detection framework. We start by describing experimental datasets and settings (Section 4.4.1 and 4.4.2), followed by detailed results analysis and case studies (Section 4.4.3 to 4.4.5).

4.4.1 Datasets

Previous studies on OOD detection mostly focus on image classification, while few have been made on natural language. Currently, there still lacks a well-established benchmark for OOD detection in NLP. Therefore, we extend the selected datasets by Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1] and propose a more extensive benchmark, where we use different pairs of NLP datasets as ID and OOD data, respectively. The criterion for dataset selection is that the OOD instances should not belong to ID classes. To ensure this, we refer to the label descriptions in datasets and manually inspect samples of instances.

We use the following datasets as alternatives of ID data that correspond to three natural language classification tasks:

- **Sentiment Analysis.** We include two datasets for this task. *SST2* [156] and *IMDB* [157] are both datasets for sentiment analysis, where the polarities of sentences are labeled either positive or negative. For SST2, the train/validation/test splits are provided in the dataset. For IMDB, we randomly sample 10% of the training instances as the validation set. Note that both datasets belong to the same task and are not considered OOD to each other.

- **Topic Classification.** We use *20 Newsgroup* [158], a dataset for topic classification containing 20 classes. We randomly divide the whole dataset into an 80/10/10 split as the train/validation/test set.

- **Question Classification.** *TREC-10* [159] classifies questions based on the types of their sought-after answers. We use its coarse version with 6 classes and randomly sample 10% of the training instances as the validation set.

Moreover, for the above three tasks, any pair of datasets for different tasks can be regarded as OOD to each other. Besides, following Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1], we also select four additional datasets solely as the OOD data: concatenations of the premises and respective hypotheses from two **NLI** datasets *RTE* [160–163] and *MNLI* [164], the English source

Dataset	# train	# dev	# test	# class
SST2	67349	872	1821	2
IMDB	22500	2500	25000	2
TREC-10	4907	545	500	6
20NG	15056	1876	1896	20
MNLI	-	-	19643	-
RTE	-	-	3000	-
Multi30K	-	-	2532	-
WMT16	-	-	2999	-

Table 4.1: Statistics of the datasets.

AUROC ↑ / FAR95 ↓		**Avg**	**SST2**	**IMDB**	**TREC-10**	**20NG**
w/o $\mathscr{L}_{\text{cont}}$	MSP	94.1 / 35.0	88.9 / 61.3	94.7 / 40.6	98.1 / 7.6	94.6 / 30.5
	Energy	94.0 / 34.7	87.7 / 63.2	93.9 / 49.5	98.0 / 10.4	96.5 / 15.8
	Maha	98.5 / 7.3	96.9 / 18.3	99.8 / 0.7	99.0 / 2.7	98.3 / 7.3
	Cosine	98.2 / 9.7	96.2 / 23.6	99.4 / 2.1	99.2 / 2.3	97.8 / 10.7
w/ $\mathscr{L}_{\text{scl}}$	$\mathscr{L}_{\text{scl}}$ + MSP	90.4 / 46.3	89.7 / 59.9	93.5 / 48.6	90.2 / 36.4	88.1 / 39.2
	$\mathscr{L}_{\text{scl}}$ + Energy	90.5 / 43.5	88.5 / 64.7	92.8 / 50.4	90.3 / 32.2	90.2 / 26.8
	$\mathscr{L}_{\text{scl}}$ + Maha	98.3 / 10.5	96.4 / 26.6	99.6 / 2.0	99.2 / 1.9	97.9 / 11.6
	$\mathscr{L}_{\text{scl}}$ + Cosine	97.7 / 13.0	95.9 / 28.2	99.2 / 4.2	99.0 / 2.4	96.8 / 17.0
w/ $\mathscr{L}_{\text{margin}}$	$\mathscr{L}_{\text{margin}}$ + MSP	93.0 / 33.7	89.7 / 49.2	93.9 / 46.3	97.6 / 6.5	90.9 / 32.6
	$\mathscr{L}_{\text{margin}}$ + Energy	93.9 / 31.0	89.6 / 48.8	93.4 / 52.1	98.4 / 4.6	94.1 / 18.6
	$\mathscr{L}_{\text{margin}}$ + Maha	99.5 / 1.7	99.9 / 0.6	100 / 0	99.3 / 0.4	98.9 / 6.0
	$\mathscr{L}_{\text{margin}}$ + Cosine	99.0 / 3.8	99.6 / 1.7	99.9 / 0.2	99.0 / 1.5	97.4 / 11.8

Table 4.2: OOD detection performance (in %) of $\text{RoBERTa}_{\text{LARGE}}$ trained on the four ID datasets. Due to space limits, for each of the four training ID dataset, we report the macro average of AUROC and FAR95 on all OOD datasets (check Appendix for full results). Results where the contrastive loss improves OOD detection on both evaluation metrics are highlighted in green ."w/o $\mathscr{L}_{\text{cont}}$+MSP" thereof is the method in Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1].

side of **Machine Translation (MT)** datasets English-German *WMT16* [165] and *Multi30K* [166]. We take the test splits in those datasets as OOD instances in testing. Particularly, for MNLI, we use both the matched and mismatched test sets. For Multi30K, we use the union of the flickr 2016 English test set, mscoco 2017 English test set, and filckr 2018 English test set as the test set. There are several reasons for not using them as ID data: (1) WMT16 and Multi30K are MT datasets and do not apply to a natural language classification problem. Therefore, we cannot train a classifier on these two datasets. (2) The instances in NLI datasets are labeled either as entailment/non-entailment for RTE or entailment/neural/contradiction for MNLI, which comprehensively covers all possible relationships of two sentences. Therefore, it is hard to determine OOD instances for NLI datasets. The statistics of the datasets are shown in Table 4.1.

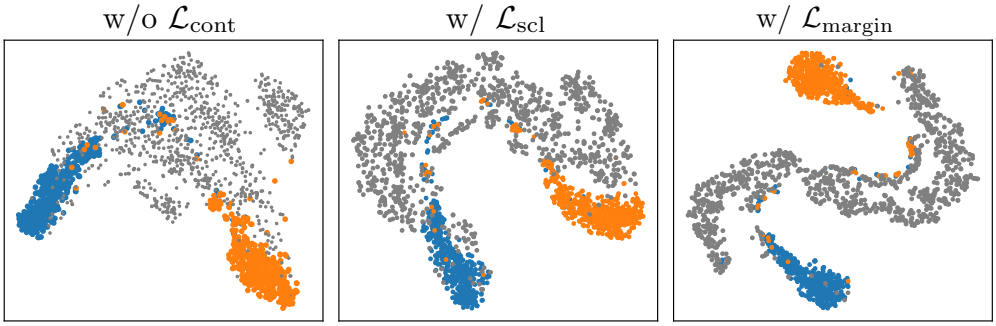

Figure 4.2: Visualization of the representations for positive, negative instances in SST2 and OOD ones. The discrepancy between ID and OOD representations is greater on representations obtained with $\mathscr{L}_{\text{margin}}$.

4.4.2 Experimental Settings

Evaluation Protocol. We train the model on the training split of each of the four aforementioned ID datasets in turn. In the inference phase, the respective test split of that dataset is used as ID test data, while all the test splits of datasets from other tasks are treated as OOD test data.

We adopt two metrics that are commonly used for measuring OOD detection performance in machine learning research [138, 143]: (1) **AUROC** is the area under the receiver operating characteristic curve, which plots the true positive rate (TPR) against the false positive rate (FPR). A *higher* AUROC value indicates better OOD detection performance, and a random guessing detector corresponds to an AUROC of 50%. (2) **FAR95** is the probability for a negative example (OOD) to be mistakenly classified as positive (ID) when the TPR is 95%, in which case a *lower* value indicates better performance. Both metrics are threshold-independent.

Compared Methods. We evaluate all configurations of contrastive losses and OOD scoring functions. Those include 12 settings composed of 3 alternative setups for contrastive losses ($\mathscr{L}_{\text{scl}}$, $\mathscr{L}_{\text{margin}}$ or w/o a contrastive loss) and 4 alternatives of OOD scoring functions (MSP, the energy score, Maha, or cosine similarity).

Model Configuration. We implement our framework upon Huggingface's Transformers [167] and build the text classifier based on RoBERTa$_{\text{LARGE}}$ [46] in the main experiment. All models are optimized with Adam [100] using a learning rate of 1e$-$5, with a linear learning rate decay towards 0. We use a batch size of 32 and fine-tune the model for 10 epochs. When training the model on each training split of a dataset, we use the respective validation split for both hyper-parameter tuning and The hyper-parameters are tuned according to the classification performance and the contrastive loss on the ID validation set. We find that $\tau = 0.3$ and $\lambda = 2$ work well with $\mathscr{L}_{\text{scl}}$, while $\lambda = 2$ work well with $\mathscr{L}_{\text{margin}}$, and we apply them to all datasets.

4.4.3 Main Results

We hereby discuss the main results of the OOD detection performance. Note that the incorporation of our OOD techniques does not lead to noticeable interference of the main-task performance, for which an analysis is later given in Section 4.4.5.

AUROC ↑ / FAR95 ↓	TREC-10	20NG
MSP	73.7 / 56.5	76.4 / 80.7
Maha	75.5 / **56.1**	77.2 / 74.1
$\mathscr{L}_{\text{margin}}$ + MSP	64.1 / 66.4	74.6 / 82.0
$\mathscr{L}_{\text{margin}}$ + Maha	**76.6** / 61.3	**78.5 / 72.7**

Table 4.3: Novel class detection performance.

The OOD detection results by different configurations of models are given in Table 4.2. For all results, we report the average of 5 runs using different random seeds. Each model configuration is reported with separate sets of results when being trained on different datasets, on top of which the macro average performance is also reported. For settings with $\mathscr{L}_{\text{scl}}$ and $\mathscr{L}_{\text{margin}}$, results better than the baselines (w/o a contrastive loss) are marked as red. We observe that: (1) Among OOD detection functions, the Mahalanobis distance performs the best on average and drastically outperforms the MSP baseline used in Hendrycks, Liu, Wallace, Dziedzic, Krishnan & Song [1]. This is due to that the Mahalanobis distance can better capture the distributional difference. (2) Considering models trained on different ID datasets, the model variants with $\mathscr{L}_{\text{margin}}$ have achieved near-perfect OOD detection performance on SST2, IMDB, and TREC-10. While on the 20 Newsgroup dataset that contains articles from multiple genres, there is still room for improvement. (3) Overall, The margin-based contrastive loss ($\mathscr{L}_{\text{margin}}$) significantly improves OOD detection performance. Particularly, it performs the best with the Mahalanobis distance, reducing the average FAR95 of Maha by 77% from 7.3% to 1.7%. (4) The supervised contrastive loss ($\mathscr{L}_{\text{scl}}$) does not effectively improve OOD detection in general. In many cases, its performance is even worse than the baseline.

4.4.4 Novel Class Detection

We further evaluate our framework in a more challenging setting of novel class detection. Given a dataset containing multiple classes (≥ 3), We randomly reserve one class as OOD data while treating others as ID data. We then train the model on the ID data and require it to identify OOD data in inference. In this case, the OOD data are sampled from the same task corpus as the ID data, and thus is much harder to be distinguished. We report the average performance of 5 trials in Table 4.3. The results are consistent with the main results in general. The Mahalanobis distance consistently outperforms consistently outperforms MSP, and the $\mathscr{L}_{\text{margin}}$ achieves better performance except for the FAR95 metric on the TREC-10 dataset. However, the performance gain is notably smaller than that in the main experiments. Moreover, none of the compared methods achieve an AUROC score of over 80%. This experiment shows that compared to detecting OOD instances from other tasks, detecting OOD instances from similar corpora is much more challenging and remains room for further investigation.

4.4.5 Analysis

Visualization of Representations. To help understand the increased OOD detection performance of our method, we visualize the penultimate layer of the Transformer trained with different contrastive

Accuracy	SST2	IMDB	TREC-10	20NG
w/o $\mathscr{L}_{\text{cont}}$	**96.4**	95.3	**97.7**	93.6
w/ $\mathscr{L}_{\text{scl}}$	96.3	95.3	97.4	93.4
w/ $\mathscr{L}_{\text{margin}}$	96.3	95.3	97.5	**93.9**

Table 4.4: Accuracy of the trained classifier.

AUROC ↑ / FAR95 ↓	L1	Cosine	L2
MSP	93.6 / 31.1	**94.1 / 30.9**	92.2 / 32.0
Energy	93.8 / 27.2	**94.7 / 26.9**	94.4 / 27.5
Maha	99.3 / 2.8	99.2 / 3.0	**99.4 / 1.7**
Cosine	98.1 / 10.9	98.8 / 5.3	**99.0 / 3.9**

Table 4.5: Average OOD detection performance of different distance metrics.

AUROC ↑ / FAR95 ↓	Maha	Maha + $\mathscr{L}_{\text{margin}}$
$\text{BERT}_{\text{BASE}}$	95.7 / 21.5	98.4 / 8.1
$\text{BERT}_{\text{LARGE}}$	97.7 / 13.3	99.1 / 3.9
$\text{RoBERTa}_{\text{BASE}}$	98.4 / 9.3	99.6 / 2.0
$\text{RoBERTa}_{\text{LARGE}}$	98.5 / 7.3	99.4 / 1.7

Table 4.6: Average OOD detection performance of other pretrained Transformers.

losses. Specifically, we train the model on SST2 and visualize instances from the SST2 validation set and OOD datasets using t-SNE [168], as shown in Figure 4.2. We observe that the representations obtained with $\mathscr{L}_{\text{margin}}$ can distinctly separate ID and OOD instances, such that ID and OOD clusters see almost no overlap.

Main Task Performance. As stated in Section 4.3.1, the increased OOD detection performance should not interfere with the classification performance on the main task. We evaluate the trained classifier on the four ID datasets. The results are shown in Table 4.4. We observe that the contrastive loss does not noticeably decrease the classification performance, nor does it increase the performance, which differs from the observations by Gunel, Du, Conneau & Stoyanov [155].

Distance Metrics. Besides L2 distance, we further evaluate the L1 distance and the cosine distance with the margin-based contrastive loss $\mathscr{L}_{\text{margin}}$. Results are shown in Table 4.5. Due to space limitations, we only report the average OOD performance on the four ID datasets. We observe that the three metrics achieve similar performance, and all outperform the baseline when using Maha as the scoring function. Among them, L2 distance gets slightly better OOD detection performance. Moreover, $\mathscr{L}_{\text{margin}}$ significantly outperforms $\mathscr{L}_{\text{scl}}$ when both use cosine as the distance metric. It shows that their performance difference arises from the characteristics of the losses instead of the metric.

OOD Detection by Other Transformers. We also evaluate the OOD detection ability of other pretrained Transformers in Table 4.6 and report the average performance on the four ID datasets. For BERT [14], we use $\lambda = 0.2$. We observe that: (1) Larger models have better OOD detection

ability. For both BERT and RoBERTa, the large versions offer better results than the base versions. (2) Pretraining on diverse data improves OOD detection. RoBERTa, which uses more pretraining corpora, outperforms BERT models. (3) The margin-based contrastive loss consistently improves OOD detection on all encoders.

4.5 Conclusion

This work presents an unsupervised OOD detection framework for pretrained Transformers requiring only ID data. We systematically investigate the combination of contrastive losses and scoring functions, the two key components in our framework. In particular, we propose a margin-based contrastive objective for learning compact representations, which, in combination with the Mahalanobis distance, achieves the best performance: near-perfect OOD detection on various tasks and datasets. We further propose novel class detection as the future challenge for OOD detection.

Chapter 5

Resolving Knowledge Conflicts

Large language models (LLMs) encode parametric knowledge about world facts and have shown remarkable performance in knowledge-driven NLP tasks. However, their reliance on parametric knowledge may cause them to overlook contextual cues, leading to incorrect predictions in context-sensitive NLP tasks (e.g., knowledge acquisition tasks). We seek to assess and enhance LLMs' contextual faithfulness in two aspects: knowledge conflict and prediction with abstention. We demonstrate that LLMs' faithfulness can be significantly improved using carefully designed prompting strategies. In particular, we identify opinion-based prompts and counterfactual demonstrations as the most effective methods. Opinion-based prompts reframe the context as a narrator's statement and inquire about the narrator's opinions, while counterfactual demonstrations use instances containing false facts to improve faithfulness in knowledge conflict situations. Neither technique requires additional training. We conduct experiments on three datasets of two standard NLP tasks, machine reading comprehension and relation extraction, and the results demonstrate significant improvement in faithfulness to contexts[1].

5.1 Introduction

Large language models (LLMs; [170–173]) have made remarkable advances in solving various NLP problems, particularly in (context-free) knowledge-driven tasks such as question answering [174, 175] and commonsense reasoning [176, 177]. Without external context, LLMs can answer factual questions and achieve comparable results to supervised approaches [170, 171], indicating that LLMs encode *parametric knowledge* about open-world facts.

Although parametric knowledge can be beneficial for knowledge-driven tasks, overly relying on it can cause problems in context-specific NLP tasks. First, LLMs may encode misconceptions [178] or obsolete facts [179–181], in which case we expect LLMs to update their predictions when provided with relevant context. Second, when using LLMs for knowledge acquisition tasks such as machine reading comprehension (MRC; [182, 183]) and information extraction (IE; [20, 85]), LLMs should always extract the *knowledge in context* instead of relying solely on their parametric knowledge. In such context-specific application scenarios, we expect LLMs to make decisions faithful to the context and avoid simply parroting answers from pretraining. However, studies have discovered that LLMs can overlook or ignore context [181, 184, 185], posing a significant challenge for their application in these scenarios.

[1]This chapter is based on Zhou, Zhang, Poon & Chen [169].

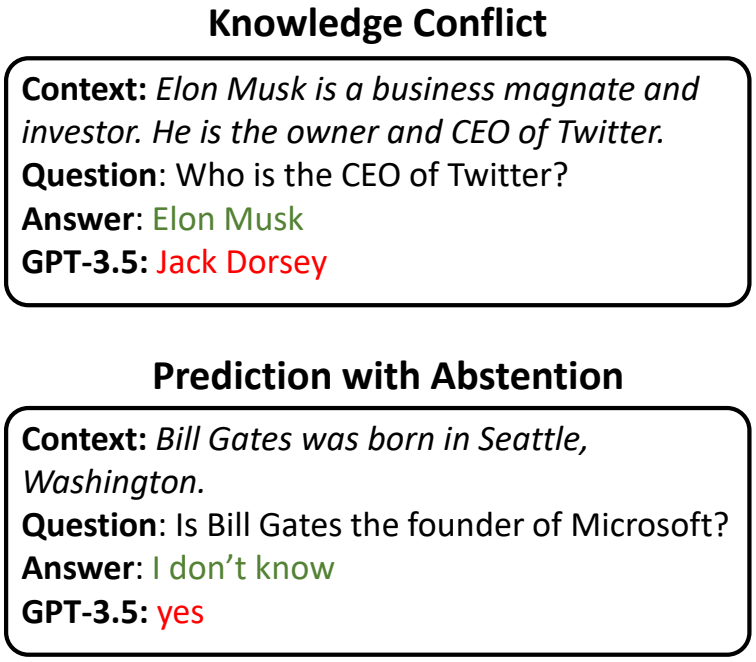

Figure 5.1: Examples of knowledge conflict and prediction with abstention. LLMs may ignore the provided context and make unfaithful predictions based on their parametric knowledge before Q4 2021.

We aim to investigate techniques for improving the faithfulness of LLMs in context-specific NLP tasks. Conceptually, faithfulness is not simply about how much accuracy the model can offer. Instead, it should concern the validity and reliability of its extraction process. Specifically, when there is a concept or relation to extract, a faithful extractor should *genuinely* induce what is described in the context but not give *trivial guesses* based on parametric knowledge or statistical biases. Besides, when no known decision-related information is described in the context, the model should *selectively* abstain from predicting. Accordingly, to provide a realistic assessment of LLMs in terms of faithfulness, we narrow our focus to two sub-problems, namely entity-based knowledge conflict [186] and prediction with abstention [187], examples of which are shown in Figure 5.1. In cases of knowledge conflict, where the given context contains facts different from the pretraining data, LLMs need to return the facts locally described in the context instead of the globally memorized facts. For example, in Figure 5.1, text-davinci-003 identifies *Jack Dorsey* instead of *Elon Musk* as the CEO of Twitter, based on its pretrained data before Q4 2021. In cases of prediction with abstention, where the provided context does not provide information to answer the questions, LLMs should abstain from making predictions and notify the users, rather than answering the questions that become a trivial guess. For example, in Figure 5.1, when asked about the founder of Microsoft based on an irrelevant context, LLMs should admit that, from here, they cannot infer the answer.

We present various prompting strategies to improve the faithfulness of LLMs, including designing effective prompts and choosing appropriate in-context demonstrations. We find that constraining the scope of questions to the context by adding phrases (e.g., based on the given context) or natural language instructions improve faithfulness in both facets. Particularly, we find that reformulating the context and questions to opinion-based question-answering problems [188, 189], where the

context is expressed in terms of a narrator's statement, and the question asks about this narrator's opinion, delivers the most gains. Additionally, we find that adding counterfactual demonstrations to prompts improves faithfulness in the aspect of knowledge conflict, while using the original (factual) demonstrations leads to limited or negative effects. Finally, combining both techniques delivers the largest gain than using each one independently.

We evaluate our methods based on three datasets, including Re-TACRED [190] for relation extraction, and natural questions [175] and RealTime QA [181] for MRC. We find that the proposed strategies can greatly improve faithfulness, e.g., reducing the memorization ratio[2] of text-davinci-003 from 35.2% to 3.0% on natural questions. Additionally, we evaluate our methods across LLMs of different scales, finding that larger LLMs are more likely to update memorized answers than smaller ones, both with and without the application of our methods.

5.2 Related Work

Knowledge conflicts. LLMs [170–172] have shown promising results in closed-book QA tasks, indicating their ability to memorize facts about the world. However, as the world is constantly evolving, memorized facts may become outdated [179–181, 191–194], emphasizing the need to update LLMs' predictions with new facts. To address this challenge, some studies [195–199] have explored ways to identify and edit the facts stored in model parameters. For example, Meng, Sharma, Andonian, Belinkov & Bau [199] identify the MLPs that have a causal effect on factual knowledge recall in LLMs and edit the parameters in these critical MLPs to update the knowledge. Their method can update up to thousands of memorized facts with little impairment to others. However, it remains unclear whether memory editing methods allow sufficient capacity to encompass all new factual knowledge. Another promising direction is to augment LLM prompting with external context containing relevant knowledge [200–202]. Coupled with retrieval systems [203–205], such methods have the potential to update LLMs with large amounts of new facts. However, such methods face the challenge that LLMs may persist with the memorized facts and ignore the provided context [186]. To tackle this challenge, recent works [184, 206] finetune LLMs on counterfactual contexts, where the original facts are replaced with counterfactual ones. They find that such finetuning processes can effectively improve the LLMs' utilization of contexts instead of relying solely on their parametric knowledge. In this study, we propose a novel approach using prompting to improve context faithfulness in LLMs without additional finetuning, which offers a more general and cost-effective method for LLMs.

Prediction with abstention. Selective prediction with abstention [207–210] is an important problem in trustworthy AI. When models are uncertain about their predictions, it is critical that they should admit the uncertainty and notify users instead of returning incorrect predictions. Selective prediction may be adopted in different scenarios, such as being on the model side where instances are close to the decision boundary [211–213], or on the data side where instances are from different domains to training [1, 128, 143]. In the scope of context-specific NLP, abstention is preferred when the context is irrelevant to the question. For example, SQuAD 2.0 [187] introduces unanswerable questions to extractive MRC, while Yatskar [214] finds it focused on questions of extreme confusion and thus is less relevant to the focus of our study. CoQA [215] and QuAC [216] introduce

[2]The percentage of times that LLMs return memorized answers versus answers in the context.

unanswerable questions to conversational question answering. RealTime QA [181] finds that GPT-3 still generates outdated answers when provided with irrelevant documents. To address the problem, Neeman, Aharoni, Honovich, Choshen, Szpektor & Abend [206] propose the answerability augmentation where LLMs should predict *Unanswerable* when presented with an empty or randomly sampled document. We tackle this problem with a part of our prompting method, which we find to significantly enhance the LLMs' ability to make selective predictions.

5.3 Method

We focus on context-specific NLP tasks. The input of these tasks is formulated as (c, q) for free-form generation tasks, where c is the context and q is the question, or (c, q, o) for tasks with close decision spaces (e.g., multi-choice tasks), where o is the set of decisions/choices. The desired output is either a free-form text or a choice. We solve these tasks by prompting LLMs and study ways of designing prompting templates and demonstrations that are dedicated to improving the faithfulness of LLMs. Specifically, we find two proposed methods, opinion-based prompts and counterfactual demonstrations, to be the most effective ones. Our methods only change the prompts without finetuning the LLMs [184, 186, 206], targeting a more general and affordable solution.

5.3.1 Opinion-based Prompting

Given an input (c, q, o), we begin with the following *base* prompting template:[3]

Base prompt

$\{c\}$ Q: $\{q\}$? Options: $\{o\}$ A:

Here, $\{\}$ serves as a placeholder that is filled with specific inputs during prompting. We investigate two types of prompting templates for context-specific NLP, namely *opinion-based* prompts and *instructed* prompts. Opinion-based prompts transform original questions into opinion-seeking questions, which naturally demand more attention to the context. Instructed prompts, on the other hand, explicitly instruct LLMs to read the context by natural language. Details of these templates are discussed in the remaining section.

Opinion-based prompts. We propose to transform the context to a narrator's statement and the question to enquire about the narrator's opinion in this statement. Our approach is motivated by our own cognitive process for answering different types of questions. When answering questions that seek factual information, we can often rely on our own memory and answer without needing to refer to the context, as these questions typically have only one correct answer. However, when questions are seeking opinions from someone else (in this context, the narrator), it is important to comprehend the narrator's words before answering the questions, as opinions may vary greatly from person to person. Besides, as opinions are inherently subjective and can be influenced by many factors such as personal experiences and beliefs, opinion-seeking questions are sometimes difficult to answer solely based on the narrator's statement compared to a fact-seeking question that typically has definite and verifiable answer(s). As a result, transforming factual questions into opinion-seeking questions can

[3] Options only apply for multiple-choice tasks and are removed in free-form text generation tasks.

lead to more attention to the context, as memorized answers alone may not suffice. It also helps the model more selectively predict under cases where contexts do not describe answers. Both factors lead to improved faithfulness on LLMs. The *opinion*-based prompting template is as follows:

Opinion-based prompt

Bob said, "{c}" Q: {q} in Bob's opinion? Options: {o} A:

Throughout our experiments, we consistently use Bob to represent the narrator for the context, although other names could be utilized as well.

Instructed prompts. We also explicitly instruct LLMs to read context by natural language. We start by extending questions in prompts with attributive phrases such as "based on the given text", leading to the following *attributed* prompting template:

Attributed prompt

{c} Q: {q} based on the given text? Options:{o} A:

We also augment the prompts with natural language instructions. Since manually writing instructions can be laborious and often fails to account for the compatibility between instructions and LLMs, we leverage automatic prompt engineering (APE; [217]) to generate the prompts. Using a few instances and their desired outputs as demonstrations, APE uses LLMs to automatically generate candidate instructions and select the best one based on the results on a dev set. We then use the following *instruction*-based prompting template:

Instruction-based prompt

Instruction: {*Instruction*} {c} Q: {q}? Options: {o} A:

Experiments show that all prompting templates perform better than the base prompting template. Specifically, opinion-based prompts outperform instructed prompts in both knowledge conflict and prediction with abstention facets, and combining these two prompting methods results in the most significant improvements.

5.3.2 Counterfactual Demonstration

Using demonstrations is a standard way to perform few-shot inference on LLMs [170]. To enhance the faithfulness of LLMs in knowledge conflict scenarios, previous works [184, 206] propose to finetune the models using counterfactual instances, where the facts in the context are substituted with false ones, and the model learns to update its predictions accordingly. Following this strategy, we propose to use counterfactual instances as demonstrations for LLMs. To do this, we start with a labeled set of counterfactual instances and a test instance and then use KATE [218] to retrieve the most relevant counterfactual instances as demonstrations. We encode both the test instance and counterfactual instances with $\text{RoBERTa}_{\text{nli+sts-b}}$ [46, 219] and select the top counterfactual instances based on cosine similarity. As a part of our analysis, we also experimented with using the original

	Method	MRC				RE			
		$p_s \uparrow$	$p_o \downarrow$	$M_R \downarrow$	EM↑	$p_s \uparrow$	$p_o \downarrow$	$M_R \downarrow$	F_1↑
Zero-shot	Base	59.0	32.1	35.2	6.2	73.9	21.5	22.5	81.0
	Attr	71.9	14.4	16.6	29.6	72.4	23.6	24.6	80.0
	Instr	74.2	16.0	17.7	27.1	75.8	15.6	17.1	81.6
	Opin	79.4	9.8	11.0	24.9	76.0	19.6	20.5	82.9
	Opin + Instr	79.1	7.9	9.1	48.6	79.4	15.0	15.9	84.7
Original	Base	43.3	49.4	53.3	35.1	76.2	19.8	20.6	83.3
	Attr	54.1	37.7	41.0	45.5	76.5	19.7	20.5	83.7
	Instr	54.6	37.7	40.8	45.8	77.3	18.4	19.2	84.2
	Opin	60.6	28.7	32.1	51.1	76.8	18.4	19.3	83.8
	Opin + Instr	64.7	26.8	29.3	53.8	78.2	17.1	17.9	84.9
Counter	Base	86.9	6.5	7.0	80.2	78.7	13.7	14.8	83.9
	Attr	89.1	4.6	4.9	83.0	79.7	13.0	14.0	84.3
	Instr	86.2	6.3	6.8	80.1	78.0	12.8	14.1	82.9
	Opin	90.1	3.7	3.9	84.3	79.7	12.8	13.8	84.4
	Opin + Instr	**90.9**	**2.8**	**3.0**	**85.2**	**80.0**	**10.5**	**11.6**	**85.1**

Table 5.1: Results (in %) in the knowledge conflict setting. The overall best results are highlighted in **bold**. The best and the second best results in each setting are highlighted in green and orange, respectively.

(factual) instances as demonstrations but found this approach to underperform counterfactual demonstrations and sometimes even zero-shot inference.

5.4 Experiments

This section presents our experimental setups (Section 5.4.1) for the evaluation of the proposed methods concerning two aspects of faithfulness: knowledge conflict (Section 5.4.2) and prediction with abstention (Section 5.4.3). We provide additional analysis (Section 5.4.4) on results across different model sizes and results on the original datasets. We also show examples of prompts and LLMs' outputs in the case study (Section 5.4.5).

5.4.1 Experimental Setup

Our experiments are conducted using the InstructGPT model (text-davinci-003, 175B parameters). We use the base prompt as our baseline, and compare it against the proposed prompting templates in Section 5.3.1, including attributed prompt (ATTR), instruction-based prompt (INSTR), opinion-based prompt (OPIN), and the combination of opinion-based prompt and instruction-based prompt (OPIN + INSTR). We evaluate the effectiveness of these templates in both zero-shot and few-shot settings (with demonstrations).

5.4.2 Knowledge Conflict

Datasets. We evaluate in the knowledge conflict setting using counterfactual datasets that contain incorrect facts, which can conflict with what the LLM has memorized. We use two datasets based on real-world texts: natural questions [175] for MRC and Re-TACRED [190] for relation extraction (RE). To create counterfactuals, we adopt the framework proposed by Longpre, Perisetla, Chen, Ramesh, DuBois & Singh [186], which modifies the context to support a counterfactual answer. Specifically, for MRC, we follow Longpre, Perisetla, Chen, Ramesh, DuBois & Singh [186] and replace the gold entity answer in the context with a randomly sampled entity of the same entity type from the corpus. For RE, we first randomly sample a context that has the entity mentions of the same type but different relations from the original one, and then insert the original entities into the sampled context. In this scenario, a faithful LLM should update its prediction to the new answer instead of returning the original one. Moreover, to measure LLMs' ability to update answers, we need to ensure that they have knowledge of the original answers in the first place. Therefore, we only evaluate LLMs on a subset of instances where they can correctly predict the original answers without additional contexts. This process left us with 2,773 test instances on natural questions and 4,353 test instances on Re-TACRED.

Task setup. We use the same set of evaluation metrics as Longpre, Perisetla, Chen, Ramesh, DuBois & Singh [186]. Specifically, we measure the frequency that the LLMs' predictions *contain* an exact match of the original answers (p_o) and the substituted answers (p_s), after both predictions and answers have been normalized by removing stop words and punctuation To assess the model's reluctance to update its prediction, we use the memorization ratio (M_R), which is calculated as $M_R = \frac{p_o}{p_o+p_s}$. A completely faithful LLM should have an M_R of 0. We also report task-specific metrics, including exact match (EM) for MRC and F_1 for RE. For EM, we also use normalized predictions and answers, but the requirement is that the prediction and answer must be exactly the same, rather than just containing the answer. We conduct experiments in three different settings: zero-shot, demonstration using original instances, and demonstration using counterfactual instances. We retrieve demonstrations from the original/counterfactual training set, and evaluate LLMs on the counterfactual test set. In the few-shot setting, we use 16 demonstration instances. Since text-davinci-003 has a maximum input length of 4,096 tokens, some instances are filtered out during evaluation.

Results and discussion. The results in Table 5.1 demonstrate that the combination of OPIN + INSTR prompting and counterfactual demonstrations is most effective. It results in a reduction of 32.2% and 10.9% in M_R for MRC and RE respectively when compared to the zero-shot base prompts. We also find that opinion-based prompts generally perform better than other templates, achieving the second-best results on 17 out of 24 metrics, indicating that LLMs are more faithful to the context when answering opinion-seeking questions. Combining opinion-based prompts and instruction-based prompts further improves faithfulness, with the best results obtained in 23 out of 24 metrics.

When it comes to few-shot settings, counterfactual demonstrations lead to further improved performance. Using the original (factual) instances as demonstrations, on the other hand, leads to limited effects or may even impair faithfulness in MRC. This finding suggests that demonstrations do not always improve the generalization of LLMs' inference, especially when they contain dataset bias. In the MRC experiments, the natural questions dataset used is constructed based on Wikipedia, which mainly consists of world facts. This potentially allows for a simplicity bias of LLMs where

	Method	Acc↑		Brier↓
		NoAns	All	All
Zero-shot	Base	30.6	68.5	29.4
	Attr	65.3	84.4	14.6
	Instr	81.6	91.7	7.7
	Opin	83.3	92.6	6.6
	Opin + Instr	87.8	94.4	5.2
Few-shot	Base	73.5	88.2	11.2
	Attr	81.6	91.9	8.0
	Instr	85.7	93.7	6.1
	Opin	87.8	94.6	4.1
	Opin + Instr	**89.8**	**95.5**	**3.4**

Table 5.2: Results (in %) on RealTime QA. The overall best results are highlighted in **bold**. The best and the second best results in each setting are highlighted in green and orange, respectively. As all prompts achieve perfect accuracy (100%) on the **HasAns** subset, it is not included in the table.

questions can be answered without contexts. Therefore, our study suggests the importance of using counterfactual demonstrations in knowledge conflict scenarios.

5.4.3 Prediction with Abstention

Datasets. As for the second aspect of faithfulness, we evaluate LLMs' ability to selectively abstain from making uncertain predictions based on irrelevant context. Since existing datasets such as SQuAD 2.0 [187] generally contain questions with confusion [214] and are less related to our problem setting, we curate our own evaluation data based on RealTime QA [181], a dataset that inquires about novel information from June 2022 onwards. In this formulation, LLMs are presented with a question and multiple choices, and they need to choose the correct answer based on several retrieved documents. These documents were obtained using tools like Google custom search and may not contain the answer to the question. To adapt this dataset to our setting, we added a new "I don't know" choice and relabeled the dataset. Instances where the retrieved documents do not answer the question are relabeled to "I don't know". We used questions in the first six weeks of 2022 as the test set and randomly picked three questions of 2023 as demonstration instances. This process results in a total of 113 test instances, including 63 answerable questions and 50 unanswerable ones.

Task setup. We calculate the probability of a choice as $P(\text{choice}|\text{prompt})$ followed by normalization across all choices.[4] We take the choice with the largest probability as the prediction. We report accuracy on the entire dataset (All), accuracy on the subset of questions that can be answered based on retrieved documents (HasAns), and accuracy on questions that cannot be answered based on retrieved documents (NoAns). The latter two metrics measure LLMs' ability to extract answers from context and their ability to abstain from making predictions when the context does not describe

[4]We tried to scale the probability of choices for length or unconditional probability as done in Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, *et al.* [170], whereas we find both lead to worse results on RealTime QA.

the answer, respectively. Besides, we use the probability of "I don't know" as LLM's probability estimation of whether the question can be answered. We use the Brier score to evaluate the accuracy of the estimation, which measures the mean squared difference between the estimation and the true binary outcome of answerability. We use three demonstrations for each instance in the few-shot setting. As text-davinci-003 has a maximum input length of 4,096 tokens, some instances are filtered out during evaluation.

Results and discussion. The results presented in Table 5.2 demonstrate that the OPIN + INSTR prompt achieves the best results in both the zero-shot and few-shot settings, outperforming base prompts by 57.2% and 16.3% in accuracy in the NoAns subset, respectively. It also reduces the Brier score by 24.2% and 7.8% compared to base prompts in the two settings, respectively. The OPIN prompt is the second best in terms of these metrics. These findings demonstrate that opinion-based prompts can enhance the LLMs' ability to make selective predictions. Furthermore, all proposed prompting templates achieve better results on NoAns instances compared to base prompts, and maintain perfect accuracy on HasAns subset, indicating their ability to improve LLMs' selective prediction without compromising performance on answerable instances. In addition, The use of demonstrations consistently improves the LLMs' ability to make selective predictions, as evidenced by the lower Brier scores in the few-shot setting compared to the zero-shot setting.

5.4.4 Additional Analysis

Memorization by different sizes of LLMs. Figure 5.2 shows the memorization ratio M_R across different sizes of InstructGPTs under the zero-shot evaluation of natural questions.[5] Overall, OPIN + INSTR consistently outperforms other prompts across different model sizes. In the upper plot, results are shown for filtered evaluation sets where the corresponding LLMs can correctly predict the original answers without additional contexts, thereof the size of evaluation sets varies across different LLMs.[6] We observe that M_R generally decreases with increased model size, showing that larger LLMs are better at updating memorized answers based on given contexts in knowledge conflicts. However, the lower plot reveals that larger LLMs have more severe memorization on the full (unfiltered) evaluation set. This is because larger LLMs can memorize more answers than smaller ones, as evidenced by the number of instances in the filtered evaluation set where larger LLMs have more instances. Our analysis suggests that while larger LLMs are better at updating memorized answers, they still tend to have more memorization due to the larger number of memorized answers. Therefore, we need to pay more attention when using larger LLMs in scenarios with new or potentially conflicting knowledge.

Selective prediction by different sizes of LLMs. Figure 5.3 shows the Brier score across different sizes of InstructGPTs under the zero-shot evaluation of RealTime QA. On smaller LLMs, opinion-based prompt achieves similar or even higher Brier score than base prompts, indicating it does not improve the selective prediction ability of LLMs. We hypothesize that this is because smaller LLMs have inferior reading comprehension ability, resulting in uncertainty in many instances. Opinion-based prompts change uncertain predictions of answerable questions to *I don't know*,

[5]The 0.3B, 1.3B, 6.7B models refer to text-ada-001, text-babbage-001, text-curie-001, respectively. We do not perform few-shot evaluation as different sizes of LLMs have different maximum input lengths and can take different numbers of demonstrations, thus hard to be compared to each other.

[6]The sizes of the filtered evaluation sets, in the order of increased model sizes, are 121, 132, 756, and 2,773.

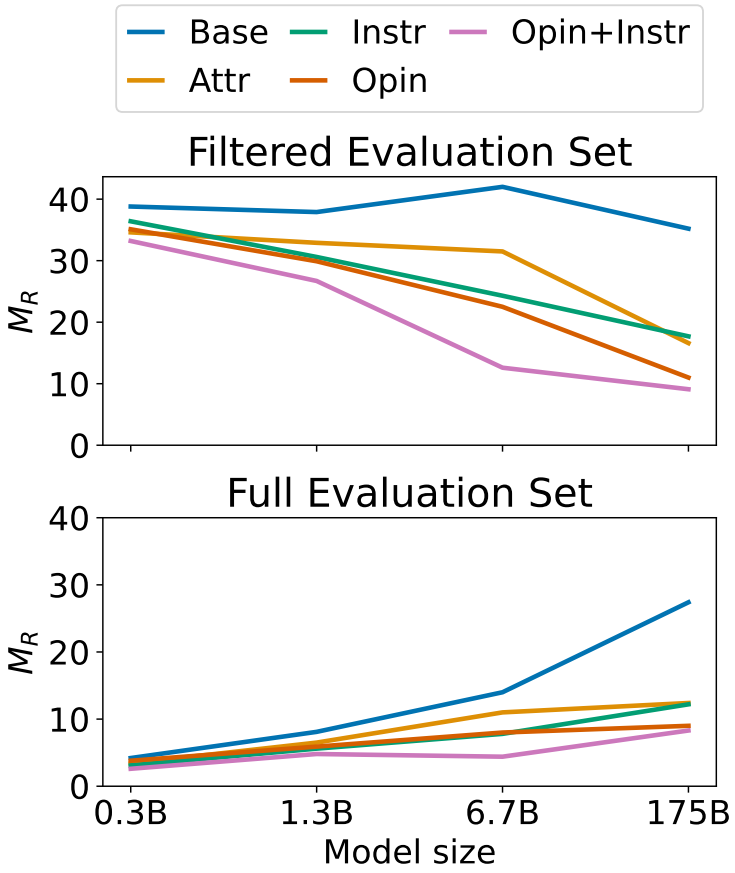

Figure 5.2: Memorization ratios across different sizes of InstructGPTs, evaluated in the zero-shot setting using natural questions.

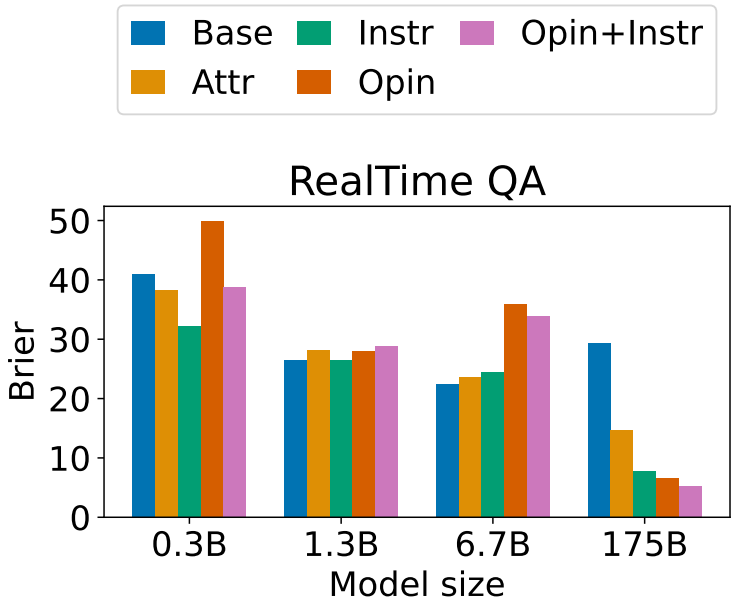

Figure 5.3: Brier scores across different sizes of InstructGPTs, evaluated in the zero-shot setting of RealTime QA.

	Method	p_o↑	EM↑
Zero-shot	Base	92.1	11.1
	Opin	91.3	25.2
	Opin + Instr	90.5	57.2
Original	Base	93.2	77.8
	Opin	92.7	78.7
	Opin + Instr	93.9	80.1
Counter	Base	93.6	82.1
	Opin	92.8	82.3
	Opin + Instr	92.7	82.1

Table 5.3: Results (in %) on the filtered evaluation set of natural questions with original (factual) contexts and answers.

which could lead to worse results. For other prompting templates, we do not observe a consistent improvement across different LLMs either. This analysis shows that while the selective prediction ability can be more easily activated by zero-shot prompting for LLMs such as text-davinci-003, smaller LLMs may require dedicated adaptations such as calibration and finetuning to activate this ability.

Results on original datasets. While our main experiments demonstrate the effectiveness of the proposed methods in resolving knowledge conflicts, LLMs in real-world applications may also see instances without knowledge conflicts. Therefore, we investigate how our methods affect

	Knowledge Conflict	Prediction with Abstention
Context	(Counterfactual passage) The Super Bowl LI Halftime show took place on February 5, 2017, at NRG Stadium in Houston, Texas as part of Super Bowl LI. The show was headlined by **Bosco**, who performed a medley of her songs, including newer material from her most recent studio album Joanne.	Tara Connolly is senior gas campaigner at Global Witness, an international NGO working towards a more sustainable, just and equal planet. She has over a decade of experience in EU energy policy. The views expressed in this commentary are her own.
Prompt	Instruction: read the given information and answer the corresponding question. Bob said, "The Super Bowl ... album Joanne." Q: who performed the halftime show at Super Bowl 51 in Bob's opinion based on the given text?	Instruction: answer a question based on the provided input-output pairs. Bob said, "Tara Connolly ... are her own." Q: Mo Farah made public that he was trafficked from which African country to the UK in Bob's opinion based on the given text? Choices: Somaliland; Djibouti; Ethiopia; Somalia; I don't know
Base	Lady Gaga ✗	Somalia ✗
Attr	Lady Gaga ✗	Somalia ✗
Instr	Lady Gaga ✗	Somaliland ✗
Opin	Bosco ✓	I don't know ✓
Instr + Opin	Bosco ✓	I don't know ✓
Answer	Bosco	I don't know

Table 5.4: Examples of prompts and LLMs' corresponding predictions. In the "Prompt" row, we show and highlight the added parts from different prompting templates including attributed prompts, instruction-based prompts, and opinion-based prompts.

inference when the memorized answers align with the given contexts. To do so, we evaluate LLMs on the same set of filtered evaluation set used in the main results section (Section 5.4.2), but we use the original contexts and answers instead of counterfactual ones. The results in Table 5.3 show that opinion-based prompts yield similar or better results in all settings. Furthermore, using either counterfactual or original demonstrations does not significantly impact results on the original (factual) dataset. This analysis reveals that our methods do not impair performance on instances without knowledge conflicts.

5.4.5 Case Study

Table 5.4 shows some examples of prompts and the corresponding answers generated by text-davinci-003. The left column of the table presents a knowledge conflict case where the original answer, *Lady Gaga*, is replaced with a counterfactual answer, *Bosco*. When using base prompts, LLM ignores the context and return the memorized answer *Lady Gaga*. However, using opinion-based prompts and their combination with instructions leads to a more faithful response, with the language model returning *Bosco* in the given context. The right column presents a scenario where the retrieved context from Google search is irrelevant to the given question. In such cases, base prompts still return a choice, leading to a potentially incorrect answer. However, opinion-based prompts and their combination with instructions can abstain from making predictions and return

I don't know. These examples demonstrate the effectiveness of proposed prompts in generating context-faithful responses.

5.5 Conclusion

We focus on addressing the faithfulness issue of LLMs in context-specific NLP tasks, particularly in scenarios with knowledge conflict and prediction with abstention. We propose that two methods, opinion-based prompts and counterfactual demonstrations, are effective in improving LLMs' faithfulness to contexts. We evaluate our methods on three datasets of two tasks, namely machine reading comprehension and relation extraction, and observed significant improvement in faithfulness to contexts. Future work includes evaluating the effectiveness of proposed methods on a broader range of NLP tasks such as open-domain QA and summarization, and studying other techniques to improve faithfulness further.

Chapter 6

Dataset Annotation by Neural Rule Grounding

Deep neural models for relation extraction tend to be less reliable when perfectly labeled data is limited, despite their success in label-sufficient scenarios. Instead of seeking more instance-level labels from human annotators, here we propose to annotate frequent surface patterns to form *labeling rules*. These rules can be automatically mined from large text corpora and generalized via a soft rule matching mechanism. Prior works use labeling rules in an exact matching fashion, which inherently limits the coverage of sentence matching and results in the low-recall issue. We present a neural approach to ground rules for RE, named NERO, which jointly learns a relation extraction module and a soft matching module. One can employ any neural relation extraction models as the instantiation for the RE module. The soft matching module learns to match rules with *semantically similar* sentences such that raw corpora can be automatically labeled and leveraged by the RE module (in a much better coverage) as augmented supervision, in addition to the exactly matched sentences. Extensive experiments and analysis on two public and widely-used datasets demonstrate the effectiveness of the proposed NERO framework, comparing with both rule-based and semi-supervised methods. Through user studies, we find that the time efficiency for a human to annotate rules and sentences are similar (0.30 vs. 0.35 min per label). In particular, NERO's performance using 270 rules is comparable to the models trained using 3,000 labeled sentences, yielding a 9.5x speedup. Moreover, NERO can predict for unseen relations at test time and provide interpretable predictions [1].

6.1 Introduction

Relation extraction (RE) plays a key role in information extraction tasks and knowledge base construction, which aims to identify the relation between two entities in a given sentence. For example, given the sentence "Bill Gates is the founder of Microsoft" and an entity pair ("Bill Gates", "Microsoft"), a relation classifier is supposed to predict the relation of `ORG:FOUNDED_BY`. Recent advance in neural language processing has shown that neural models [20, 30, 220] gained great success on this task, yielding state-of-the-art performance when a large amount of well-annotated sentences are available. However, these supervised learning methods degrade dramatically when the sentence-level labels are insufficient. This problem is partially solved by the use of distant supervision based on knowledge bases (KBs) [221, 222]. They usually utilize an existing KB for automatically annotating an unlabeled corpus with an over-simplified assumption — two entities

[1] This chapter is based on Zhou, Lin, Lin, Wang, Du, Neves, *et al.* [69].

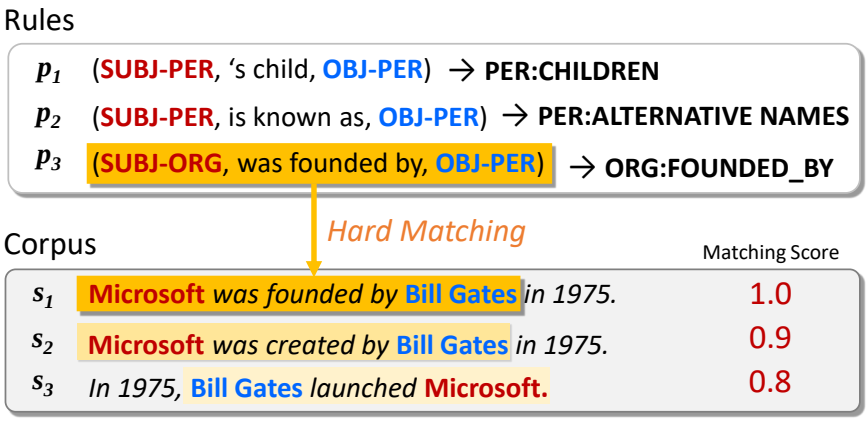

Figure 6.1: Current rule-based methods mostly rely on exact/hard matching to raw corpus and suffer from limited coverage. For example, the rule body of p_3 only matches sentence s_1 but is also similar to s_2 and s_3, which express the same relation as p_3. A "soft rule matching" mechanism is desirable to make better use of the corpus for label generation.

co-occurring in a sentence should be labeled as their relations in the KB regardless of their contexts. While distant supervision automates the labeling process, it also introduces noisy labels due to context-agnostic labeling, which can hardly be eliminated.

In addition to KBs, labeling rules are also important means of representing domain knowledge [89], which can be automatically mined from large corpora [223, 224]. Labeling rules can be seen as a set of pattern-based heuristics for matching a sentence to a relation. These rules are much more accurate than using KBs as distant supervision. The traditional way of using such rules is to perform exact string matching, and a sentence is either able or unable to be matched by a rule. This kind of hard-matching method inherently limits the generalization of the rules for sentences with similar semantics but dissimilar words, which consequently causes the low-recall problem and data-insufficiency for training neural models. For example, in Fig. 6.1, the rule P_3 can only find S_1 as one hard-matched instance for training the model while other sentences (e.g., S_2 and S_3) should also be matched for they have a similar meaning.

Many attempts have been made to RE using labeling rules and unlabeled corpora, as summarized in Fig. 6.2. *Rule-based bootstrapping* methods [225–227] extract relation instances from the raw corpus by a pre-defined rule mining function (e.g., TF-IDF, CBOW) and expanding the rule set in an iterative way to increase the coverage. However, they still make predictions by performing hard-matching on rules, which is context-agnostic and suffers from the low-recall problem. Also, their matching function is not learnable and thus has trouble in extracting semantically similar sentences. Another typical option is *data programming* [89, 228], which aims to annotate the corpus using rules with model fitting. It trains a neural RE model using the hard-matched sentences by rules and then reduces the noise of rules with their proposed algorithms. However, it does not consider the massive data that fail to be annotated by hard-matching. *Self-training* [229], as a semi-supervised framework, does attempt to utilize unmatched sentences by using confident predictions of a learned model. Then, they train the model again and again by iteratively generating more confident predictions. However, it does not explicitly model the soft matching ability of rules over unmatched sentences, making the generated labels noisy and unreliable.

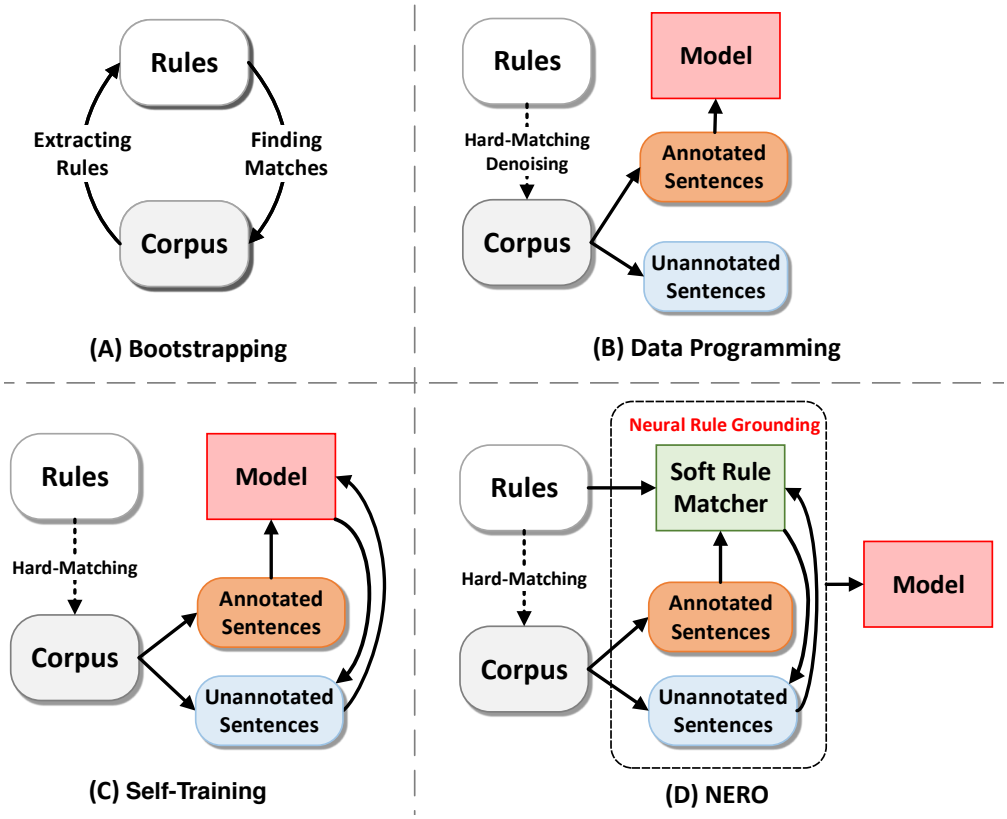

Figure 6.2: Comparison between previous work and the proposed NERO framework. (A) Bootstrapping. (B) Data Programming. (C) Self-Training. (D) NERO. The neural rule grounding process enables the soft-matching between rules and unmatched sentences using the soft rule matcher.

We want to explicitly exploit labeling rules over unmatched sentences as supervision for training better RE models, for which we propose a NEural Rule grOunding (NERO) framework. The NERO framework has two major modules: a sentence-level relation classifier and a soft rule matcher. The former aims to learn the neural representations of sentences and classify which relation it talks about, which serves as the outcome of NERO. We first apply our collected rules on a raw corpus by *hard matching* and use hard-matched sentences as the main training data for NERO. The unmatched sentences are assigned with pseudo labels by the soft rule matcher, which is a *learnable* module that produces matching scores for unmatched sentences with collected rules. The key intuition behind our soft rule matcher is that the distances between rules and sentences can be modeled by simple cosine computations in a new space transformed from their neural representations (e.g. word embeddings), which can be learned by a contrastive loss. Jointly training the two modules reinforce the quality of pseudo-labels and then improve the performance of the relation classification module. Specifically, in a batch-mode training process, we use the up-to-date learned soft matcher to assign all unmatched sentences with weighted labels, which further serve as an auxiliary learning objective for the relation classifier module.

Extensive experiments on two public datasets, TACRED [20] and Semeval [230], demonstrate that the proposed NERO consistently outperforms baseline methods using the same resources by a large margin. To investigate the label efficiency of NERO, we further conduct a user study to compare models trained with rules and labeled sentences, respectively, both created under the same time constraint. Results show that to achieve similar performance, NERO requires about ten times fewer human efforts in annotation time than traditional methods of gathering labels.

The **contributions** of our work are summarized as follows: (1) Our proposed method, NERO, is among the first methods that can learn to generalize labeling rules for training an effective relation extraction model, without relying on additional sentence-level labeled data; (2) We propose a learnable rule matcher that can semantically ground rules for sentences with similar meaning that are not able to be hard-matched. This method also provides interpretable predictions and supports few-shot learning ability on unseen relations; (3) We conduct extensive experiments to demonstrate the effectiveness of our framework in a low-resource setting and show careful investigations on the efficiency of using human efforts.

6.2 Problem Formulation

We first introduce basic concepts and their notations and then present the problem definition as well as the scope of the work.

Relation Extraction. Given a pair of entity strings (e_{subj}, e_{obj}), identified as the `subject` and `object` entities respectively, the task of relation extraction (RE) is to predict (classify) the relation $r \in \mathscr{R} \cup \{\text{NONE}\}$ between them, where $\mathscr{R}$ is a pre-defined set of relations of interest. Specifically, here we focus on *sentence-level* RE [20, 230, 231], which aims to predict the relation of entity mention pairs in a sentence s—*i.e.*, identifying the label $r \in \mathscr{R}$ for an *instance* in the form of $(e_{\text{subj}}, e_{\text{obj}}; s)$[2] and yielding the prediction $(e_{\text{subj}}, r, e_{\text{obj}}; s)$.

Labeling Rules. As an alternative to instance-level human-annotated data (*e.g.*, instance-label pairs), labeling rules formalize human's domain knowledge in a structured way, and can be either directly

[2]We use "instance" and "sentence" exchangeably in the rest of the chapter.

used for making predictions over test instances [226], or applied to generate labeled instances for model learning [89]. Formally, an inductive labeling rule consists of a *rule body* and a *rule head*. In the context of relation extraction, we consider the cases where the rule body is a textual pattern $p = [\text{SUBJ-TYPE}; c; \text{OBJ-TYPE}]$, in which c denotes a word sequence between the two entities in a sentence (*i.e.*, context), and SUBJ-TYPE and OBJ-TYPE specify the entity types of the subject and object entities required by the rule body. An instance $(e_{\text{subj}}, e_{\text{obj}}; s)$ is *hard-matched* by a rule p if and only if p's context c exactly match the context between e_{subj} and e_{obj} in s and the entity types of e_{subj} and e_{obj} match p's associated entity types. For example, sentence s_1 in Fig. 6.1 is hard-matched by rule p_3.

Such labeling rules can either be hand-written by domain experts, or automatically generated from text corpora with a knowledge base that shares the relations of interests. In this work, we adopt a hybrid approach (as detailed in Sec. 6.3.2): 1) we first extract surface patterns (*e.g.* frequent patterns, relational phrases, *etc.*) as candidate rule bodies from raw corpora (after running named entity recognition tool) using pattern mining tools, and 2) ask human annotators to assign relation labels (*i.e.*, rule heads) to the rule bodies. Then, we apply these human-annotated rules over instances from a raw corpus to generate labeled instances (as training data). Compared to directly annotating instances, our study will show that annotating rules is much more label-efficient, as one rule may match many instances in the corpus with reliable confidence, generating multiple labeled instances. Moreover, annotating rules may require similar (or even less) time than annotating instances since they are shorter in length, as shown in our later analysis.

Example 1 (Labeling Rule Generation) *Given instances such as* {*(Bill Gates, Microsoft; "Bill Gates founded Microsoft."), (Bill Gates, Steve Jobs; "Guests include Bill Gates and Steve Jobs.")*}*, the candidate patterns include* {"SUBJ-PERSON *founded* OBJ-PERSON"*,* "SUBJ-PERSON*,* OBJ-PERSON"} *, and the rule set after human annotation includes* {"SUBJ-PERSON *founded* OBJ-PERSON" $\rightarrow$ FOUNDED_BY}*. The second pattern is filtered out by annotators due to its low quality.*

Problem Definition. We focus on sentence-level relation extraction using labeling rules created by a semi-automatic method. Specifically, given a raw corpus consisting of a set of instances $\mathscr{S} = \{(e^i_{\text{subj}}, e^i_{\text{obj}}; s^i)\}_{i=1}^N$, our approach first extracts surface patterns $\{[\text{SUBJ-TYPE}; c; \text{OBJ-TYPE}]\}$ as candidate rule bodies, and then annotates them to generate labeling rules $\mathscr{P} = \{p^k\}_{k=1}^{|\mathscr{P}|}$ with p^k denoting $[\text{SUBJ-TYPE}^k; c^k; \text{OBJ-TYPE}^k] \rightarrow r_k$ and $r_k \in \mathscr{R} \cup \{\text{NONE}\}$. Next, our framework aims to leverage both the rules $\mathscr{P}$ and the corpus $\mathscr{S}$ to learn an effective relation classifier $f : \mathscr{S} \rightarrow \mathscr{R} \cup \{\text{NONE}\}$ which can predict relation for new instances in unseen sentences.

In addition to predicting relations already observed in the training phase, we aim to **predict unseen relations** solely based on new rules specified for the unseen relations at the test time—*i.e.*, a few-shot learning scenario. Such a generalization ability is desirable for domain adaptation. In this setting, the model is trained on a corpus $\mathscr{S}$ using labeling rules defined on relation set $\mathscr{R}_c$. In the testing phase, the model is expected to make predictions on unseen relation types $\mathscr{R}_u$ given some corresponding labeling rules.

Our Focus. Instead of relying on pre-existing knowledge bases to conduct distant supervision, we focus on the scenarios where labeling rules are relatively cheaper and faster to obtain by automatically inducing from large text corpora (as compared to manually curating a KB from scratch). We study neural rule grounding to integrate information from rules and corpora into

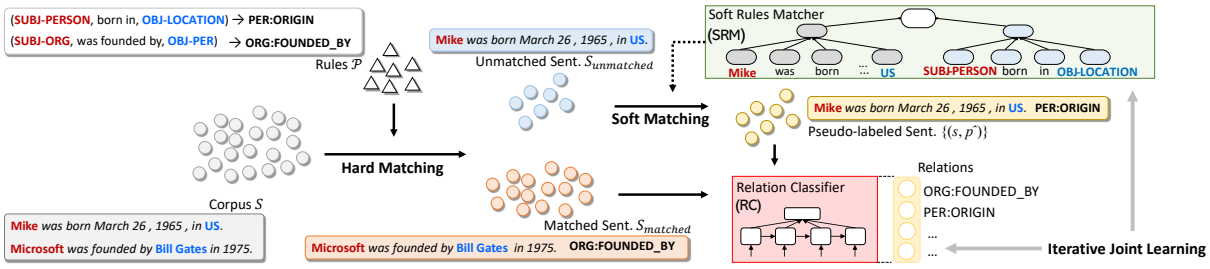

Figure 6.3: Overview of the NERO framework. Each unmatched sentence is first annotated by the soft rule matcher (SRM) to generate pseudo labels, and then fed into the relation classifier (RC) to update the model parameters. The whole framework is trained iteratively and jointly, based on multiple loss functions as introduced in Sec. 6.3.5.

learning an effective relation extraction model that can generalize better than solely using rules in a hard-matching manner. We mainly use surface pattern rules (*e.g.*, similar to the ones used in [232]) and leave the study of more complex rules (*e.g.*, regular expression) as future work.

6.3 Neural Rule Grounding (NERO)

This section introduces the important concepts in NERO, and provides details of the framework design and learning objectives. We first present our key motivation, then give an overview of NERO framework (Sec. 6.3.1) and introduce the details of each component (Secs. 6.3.2-6.3.4). Lastly, we show how we jointly learn model parameters in NERO for sequence encoding and rule matching (Sec. 6.3.5).

Motivation of Soft Rule Matching. Traditional methods of using labeling rules for relation extraction are mainly based on exact string/pattern matching (*i.e.*, *hard-matching*), where a sentence can be annotated by a rule if and only if it has exactly the same surface form as the rule body. While being easy to implement and yielding relatively high precision for the prediction, these hard-matching methods only achieve sub-optimal performance because of the severe low recall issue (see Fig. 6.1 for an example). Such a low coverage of using rules can only produce a minimal number of "labeled" sentences for learning a RE model, which may be far from enough for data-hungry neural models. To address this problem, we would like to study "soft-matching" for exploiting the rules for relation extraction. A soft-rule matcher is proposed to semantically match a rule even when the surfaces are different (e.g. "founded" v.s. "created"). Towards improving relation extraction with soft-matching, we would like to learn a SRM such that we can assign *pseudo-label*s to sentences. Further, we can use these pseudo-labeled sentences to learn a sentence-level relation classifier.

6.3.1 Framework Overview

The NERO framework consists of a relation classifier (RC) and a soft rule matcher (SRM). The RC is for learning neural representations for sequences of words, from either an instance or a rule body (Sec. 6.3.3). The SRM gives a matching score to indicate the similarity between a rule body and an instance (Sec. 6.3.4). Previous works [227] mostly adopt a *fixed* metric function (*e.g.*, cosine similarity measure between the sequence embeddings). In contrast, our SRM is a *learnable* neural

network and thus can be more effective in modeling the semantic similarity between sentences and rule bodies. As shown in Fig. 6.3, we first apply our collected rules to a raw corpus $\mathscr{S}$ to perform *hard matching*. $\mathscr{S}$ will then be divided into two subsets: "*hard-matched sentences*" $\mathscr{S}_{\text{matched}}$ and "*unmatched sentences*" $\mathscr{S}_{\text{unmatched}}$. Lastly, we use SRM and rules to iteratively generate pseudo labels over $\mathscr{S}_{\text{unmatched}}$, while jointly learning RC and SRM with the pseudo labels to promote a common representation space (Sec. 6.3.5).

6.3.2 Labeling Rule Generation

As introduced in Sec. 6.2, our candidate rule bodies (*i.e.*, surface patterns) are automatically extracted from the raw corpus. In this work, we adopt a simple yet effective pattern mining method. Given a raw corpus, we first replace entities with entity type masks SUBJ/OBJ-NER, where "NER" denotes entity type (same as the procedure in [20]). Then we pick the word sequences between and including the two entities as candidate rules. So each candidate rule is just a short phrase / sentence containing two masked entities. To reduce the annotation efforts and ensure the popularity of rules, we convert all words to their root form using Porter stemming algorithm [233] and only keep the rules whose stemmed form appears at least N times in the whole corpus. Finally, we ask human annotators to select rules that indicate a relation and assign labels to them. This process is similar to label a sentence with a pre-defined relation or NONE. In spite of the existence of more complex rule extraction methods such as shortest dependency path [223, 234] and meta patterns [224], we adopt frequent word sequences in this work because of their simplicity and high readability. Our framework can also be adapted to other rule generation methods as long as they can generate human-readable candidate rules. Regardless of their high generalization ability, labeling rules may bias the label distribution, *e.g.* PER:TITLE counts for 19% in the official training data while only counts for 5.5% in the matched sentences. This domain gap poses another challenge for learning with rules.

6.3.3 Relation Classifier (RC)

The relation classifier (named as RC) is the end product of our NERO framework, which can represent a relation instance (e_1, e_2, s) into vector embeddings. This module is general and can use various designs (*e.g.*, attention CNN [235], attention RNN [20, 236], and GCN [30, 37]). In this work, we stick with the LSTM + ATT model [236]. Given a sequence of n words, we first look up their word embeddings as $\{\mathbf{x_t}\}_{t=1}^{n}$. Then we use a bi-directional LSTM network [38] to obtain the contextualized embeddings of each word $\{\mathbf{h_t}\}_{t=1}^{n}$ where $\mathbf{h_t} \in \mathbb{R}^{d_h}$. Finally, we employ an attention layer [237] to get a sentence representation $\mathbf{c}$:

$$\{\mathbf{h_t}\}_{t=1}^{n} = \text{BiLSTM}(\{\mathbf{x_t}\}_{t=1}^{n}) \tag{6.1}$$

$$\alpha_t = \frac{\exp(\mathbf{v}^{\mathrm{T}} \tanh(\mathbf{A}\mathbf{h_t}))}{\sum_{t'=1}^{n} \exp(\mathbf{v}^{\mathrm{T}} \tanh(\mathbf{A}\mathbf{h_{t'}}))} \tag{6.2}$$

$$\mathbf{c} = \sum_{t=1}^{n} \alpha_t \mathbf{h}_t \tag{6.3}$$

where $\mathbf{A} \in \mathbb{R}^{d_a \times d_h}$, $\mathbf{v} \in \mathbb{R}^{d_a}$ are learnable model parameters for the attention mechanism. We can get the relation type probability distribution by feeding $\mathbf{W}_{\text{rc}}\mathbf{c}$ into a SoftMax layer:

$$\texttt{RC}(s, e_{\text{subj}}, e_{\text{obj}}) = \text{SoftMax}(\mathbf{W}_{\text{rc}}\mathbf{c}),$$

where $\mathbf{W}_{\text{rc}} \in \mathbb{R}^{d_h \times |\mathscr{R}|}$ is a matrix to learn. In other words, we are using the output vector of the *RC* for modeling the conditional distribution of relation $\mathbb{P}_{\theta_{RC}}(r = i|s) = \texttt{RC}(s, e_{\text{subj}}, e_{\text{obj}})[i]$, meaning that $i-th$ element of the RC output vector as the probability of the i-th relation for the input sentence s. The θ_{rc} denotes the set of model parameters of RC that we need to learn, including $\mathbf{A}, \mathbf{v}, \mathbf{W}_{\text{rc}}$, and the weights of the BiLSTM.

6.3.4 Soft Rule Matcher (SRM)

The core component of the NERO is the soft rule matcher (named to be SRM), which is defined as a neural function for modeling the matching score between a sentence s and a rule pattern p. We formulate the Soft Rule Matcher as a function $\texttt{SRM} : (s, p) \to [-1, 1]$. As shown in Fig. 6.4, we first map the sentence s and rule p into the same embedding space by the RC, and then apply a distance metric on their embeddings to get the matching score ranging from -1 to 1. Note that a matching score of 1 indicates the sentence can be hard-matched by the rule. Because the rules are typically short phrases, we use word-level attention – a weighted continuous bag-of-words model to calculate matching scores. We explored more designs in later experiments and this part can be further studied in future work. We represent a sentence and a rule pattern as the sequence of the word embeddings, $\{\mathbf{x}^{\mathbf{s}}_{\mathbf{t}}\}_{t=1}^{n}$ and $\{\mathbf{x}^{\mathbf{p}}_{\mathbf{t}}\}_{t=1}^{m}$, respectively. By applying the attention on word embeddings instead of contextualized embedding generated by LSTM, we aim to reduce over-fitting given the scarcity of rules per relation. Note that we are using a different set of parameters for this word embedding-level attention ($\mathbf{B}$ and $\mathbf{v}$). Specifically, we have

$$\mathbf{z_s} = \sum_{t=1}^{n} \frac{\exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x}^{\mathbf{s}}_{\mathbf{t}}))}{\sum_{t'=1}^{n} \exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x}^{\mathbf{s}}_{\mathbf{t}'}))} \mathbf{x}^{\mathbf{s}}_{\mathbf{t}}, \tag{6.4}$$

$$\mathbf{z_p} = \sum_{t=1}^{m} \frac{\exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x}^{\mathbf{p}}_{\mathbf{t}}))}{\sum_{t'=1}^{m} \exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x}^{\mathbf{p}}_{\mathbf{t}'}))} \mathbf{x}^{\mathbf{p}}_{\mathbf{t}}, \tag{6.5}$$

$$\texttt{SRM}(s, p) = \frac{(\mathbf{D}\mathbf{z_s})^T(\mathbf{D}\mathbf{z_p})}{\|\mathbf{D}\mathbf{z_s}\|\|\mathbf{D}\mathbf{z_p}\|}, \tag{6.6}$$

where $\mathbf{z_s}$, $\mathbf{z_p}$ are the representations for the sentence s and the rule p respectively, and $\mathbf{D}$ is a trainable diagonal matrix which denotes the importance of each dimension. Similar idea has been shown effective in other tasks for capturing semantically close phrases, and it generalizes well when only limited training data is available [238]. In practice, a sentence may be very long and contain much irrelevant information. So for the sentence, we only keep the words between (and including) the subject and object entities, which is a common practice in other rule-based RE systems [227, 234].

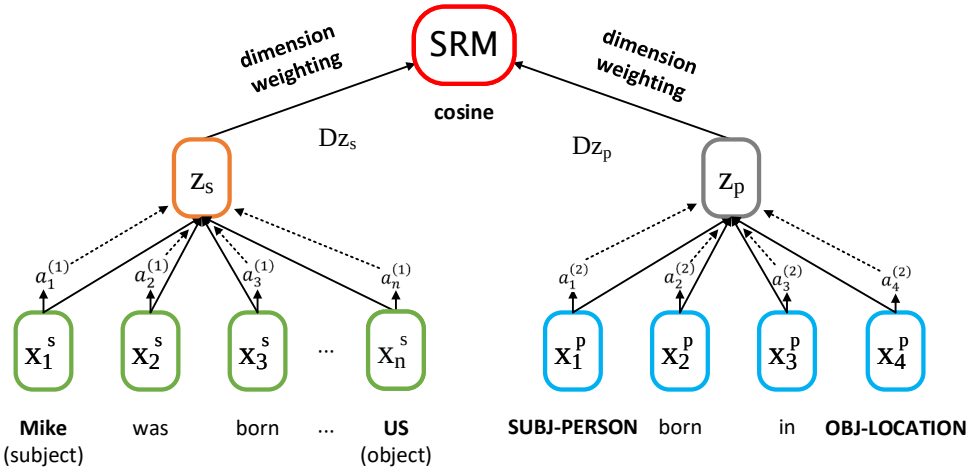

Figure 6.4: Detailed architecture of the soft rule matcher (SRM). The cosine similarity between two embeddings indicates the degree of matching between rules and sentences.

6.3.5 Joint Module Learning

We now formulate the overall learning process of NERO in a low-resource learning scenario, where we only have collected rules but no sentence-level labels at all. Given a raw corpus $\mathscr{S}$ (i.e. a large set of sentences without human-annotated labels) and a set of inductive labeling rules $\mathscr{P}$, we first exactly match every rule on $\mathscr{S}$ and get a set of **hard-matched sentences** named $\mathscr{S}_{\text{matched}}$. These hard-matched labels are pretty accurate[3], which can serve the purpose as training examples. The other **unmatched sentences**, denoted as $\mathscr{S}_{\text{unmatched}}$ are also informative for our model, while they are ignored by most prior works. Although not able to be exactly matched by any rules, the unmatched sentences can semantically align the meaning of some rules and thus help improve the RE performance. By applying the SRM upon them to create pseudo-labels, we can further utilize the hidden useful information as the supervision for learning the final RE model. Apart from that, the SRM is also optimized for improving the quality of matching in the meantime. Finally, supervision signals from $\mathscr{S}_{\text{matched}}$, $\mathscr{S}_{\text{unmatched}}$, and $\mathscr{P}$ together train the RE model in a joint learning schema.

Learning with Hard-Matched Sentences ($\mathscr{S}_{\text{matched}}$). As the primary goal of the learning, we want to minimize the error of the RC in classifying relations in the sentences in $\mathscr{S}_{\text{matched}}$ respect to their matched relation labels. Given a hard-matched sentence $s \in \mathscr{S}_{\text{matched}}$ and its associated relation $r_s \in \mathscr{R}$, we aim to minimize the cross-entropy loss L_{matched} as follows.

$$L_{\text{matched}}(\theta_{RC}) = \mathbb{E}_{s \sim \mathscr{S}_{\text{matched}}} \left[-\log \mathbb{P}_{\theta_{RC}}(r = r_s | s) \right]. \tag{6.7}$$

Learning with Rules ($\mathscr{P}$). To incorporate our collected rules for exploiting the unmatched sentences $\mathscr{S}_{\text{unmatched}}$ as well, we propose two additional auxiliary tasks for imposing rules in the RC and SRM. First, we can treat the rule body $p \in \mathscr{P}$ as a "sentence" and its rule head r_p as its associated label, thus forming a labeled instance (p, r_p). With such rule-reformatted instances as

[3]We conduct an experimental analysis on the quality of these matches in Sec. 6.5.

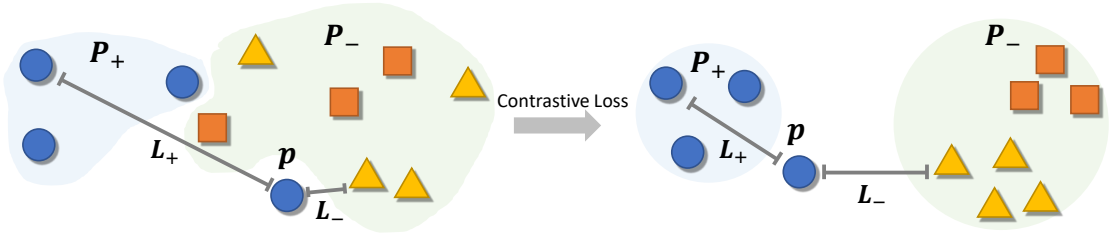

Figure 6.5: The contrastive loss in learning the SRM, which increases the matching score of rules with the same relation type and decrease the matching score otherwise.

training data, we aim to minimize the error for RC in classifying them as follows. This objective helps RC to explicitly reinforce the memory of RC about the collected rules.

$$L_{\text{rules}}(\theta_{RC}) = \mathbb{E}_{p\sim\mathscr{P}}\left[-\log \mathbb{P}_{\theta_{RC}}(r = r_p|p)\right] \tag{6.8}$$

More importantly, we present how we utilize the rules for learning the soft rule matcher (SRM) such that we can exploit the unmatched sentences as distant supervision. Our key motivation here is that a good matcher should be able to clustering rules with the same type, and thus we expect the SRM to increase the distances between rules with different types and reduce the variance of the distances between rules with the same types. Simply put, given two rules, their similarity score should be high if they belong to the same relation type and low otherwise.

We use the contrastive loss [239] for this objective. Given a rule $p \in \mathscr{P}$ and its relation $r_p \in \mathscr{R}$, we divide all the other rules with the same relation type as $\mathscr{P}_+(p) = \{p'|r_{p'} = r_p\}$, and the ones with different types as $\mathscr{P}_-(p) = \{p'|r_{p'} \neq r_p\}$. The contrastive loss aims to pull the rule p closer to its most dissimilar rule in $\mathscr{P}_+(p)$ and in the meantime push it away from its most similar rule in $\mathscr{P}_-(p)$. Mathematically, the loss is defined as follows:

$$L_{\text{clus}} = \mathbb{E}_{p\sim\mathscr{P}}\left[\max_{p_i\in\mathscr{P}_+(p)} \text{dist}_+(p,p_i) - \min_{p_j\in\mathscr{P}_-(p)} \text{dist}_-(p,p_j)\right], \tag{6.9}$$

where the measurement of the most dissimilar same-type rule and most similar different-type rule can be measured with the distances defined as below:

$$\text{dist}_+(p,p_i) = \max\left(\tau - \text{SRM}(p,p_i),\ 0\right)^2,$$
$$\text{dist}_-(p,p_j) = 1 - \max\left(\text{SRM}(p,p_j),\ 0\right)^2.$$

τ is a hyper-parameter for avoiding collapse of the rules' representations. Minimizing L_{clus} pushes the matching scores of rules that are of the same relation type up to τ and pull the matching score down to 0 otherwise. Without labels about the alignment between rules $\mathscr{P}$ and unmatched sentences $\mathscr{S}_{\text{unmatched}}$, this objective can be seen as a secondary supervision for SRM to train the parameters θ_{SRM} using $\mathscr{P}$ itself only by this contrastive loss.

Learning with Unmatched Sentences ($\mathscr{S}_{\text{unmatched}}$). We use the SRM to label unmatched sentences $\mathscr{S}_{\text{unmatched}}$ (see Fig. 6.3). As the labels generated by the SRM may be noisy, we propose to use pseudo-labeling [240] with instance weighting [241] to alleviate the noise. Each unmatched sentence is weighed by the matching score from the SRM. Specifically, for each unmatched sentence

$s \in \mathscr{S}_{\text{unmatched}}$, we first apply the SRM to compute its matching scores with each rule $p \in \mathscr{P}$ and then assign the pseudo-label to the sentence s as the relation of the highest-score rule $r_{\hat{p}}$, where

$$\hat{p} = \arg\max_{p \in \mathscr{P}} \texttt{SRM}(s, p).$$

The corresponding weight for a particular pseudo-labeled instance, $(s_i, \hat{p}_i)$, is the SoftMax over all the sentences (in a mini-batch when training; see Sec. 6.4.1 for more details):

$$w_s = \frac{\exp(\sigma \texttt{SRM}(s, \hat{p}_i))}{\sum_{s' \in \mathscr{S}_{\text{unmatched}}} \exp(\sigma \texttt{SRM}(s', \hat{p}_j))}, \tag{6.10}$$

where σ represents the temperature of the SoftMax function. The loss function for each batch of unmatched sentences would be:

$$L_{\text{unmatched}}(\theta_{RC}) = \mathbb{E}_{s \sim \mathscr{S}_{\text{unmatched}}} \left[-w_s \log \mathbb{P}_{\theta_{RC}}(r = r_{\hat{p}} | s) \right]. \tag{6.11}$$

Joint Optimization Objective. The whole framework is jointly trained under the overall loss function:

$$L(\theta_{\text{RC}}, \theta_{\text{SRM}}) = L_{\text{matched}}(\theta_{\text{RC}}) + \alpha \cdot L_{\text{rules}}(\theta_{\text{RC}}) + \beta \cdot L_{\text{clus}}(\theta_{\text{SRM}}) + \gamma \cdot L_{\text{unmatched}}(\theta_{\text{RC}}).$$

To sum up, the proposed NERO framework has two trainable components: a relation classifier (RC) and a soft rule matcher (SRM) with their specific parameters (θ_{RC} and θ_{SRM}). Our primary task is to minimize the error of the RC on the hard-matched sentences (L_{matched}). To exploit the unmatched sentences with collected rules, we first let RC to explicitly learn the rules with the objective of rule classification (L_{rules}) and also learn the SRM with the help of contrastive loss (L_{clus}) for clustering rules. We jointly learn the two modules by connecting them through pseudo-labeling on the unmatched sentences with the SRM and then expect the RC has better performance on such pseudo-labeled unmatched sentences as well ($L_{\text{unmatched}}$).

6.4 Model Learning and Inference

In this section, we introduce the specific training details of our proposed framework, and how the model conduct inference.

6.4.1 Parameter Learning of NERO

NERO starts with a raw corpus $\mathscr{S}$ and pre-defined relations $\mathscr{R} \cup \{\text{NONE}\}$. It first extracts candidate rules from $\mathscr{S}$ with pattern mining tools, then asks human annotators to select and label the rules to get the rule set $\mathscr{P}$. Before training, NERO does a quick pass of $\mathscr{S}$ by hard-matching to split it into hard-matched sentences $\mathscr{S}_{\text{matched}}$ and unmatched sentences $\mathscr{S}_{\text{unmatched}}$. Both of them are utilized to train the relation classifier RC in a joint-training manner. Algorithm 3 summarizes the overall training procedure.

The joint training objective is efficiently optimized by batch-mode training. Specifically, in each training step we randomly sample a batch of hard-matched sentences $\mathscr{B}_{\text{m}}$ from $\mathscr{S}_{\text{matched}}$, and then

Algorithm 3: Optimization of NERO model

Input: A raw corpus $\mathscr{S}$, pre-defined relations $\mathscr{R} \cup \{\text{NONE}\}$.
Output: A relation classifier $f : \mathscr{S} \to \mathscr{R} \cup \{\text{NONE}\}$.
Extract candidate rules from $\mathscr{S}$ with pattern mining tools.
Ask human annotators to select and label the candidate rules to get $\mathscr{P}$.
Partition $\mathscr{S}$ into $\mathscr{S}_{\text{matched}}$ and $\mathscr{S}_{\text{unmatched}}$ by hard-matching with $\mathscr{P}$.
while *L in* Eq. *6.3.5 not converge* **do**
  Sample batch $\mathscr{B}_m = \{(s_i, r_i)\}_{i=1}^{n}$ from $\mathscr{S}_{\text{matched}}$.
  Update L_{matched} by *Eq.* 6.7.
  Sample batch $\mathscr{B}_u = \{s_j\}_{j=1}^{m}$ from $\mathscr{S}_{\text{unmatched}}$.
  foreach $s \in \mathscr{B}_u$ **do**
    Find highest-scored rule $\hat{p}$ and pseudo label $r_{\hat{p}}$ by SRM.
  Update $L_{\text{unmatched}}$ by *Eq.* 6.12.
  Update L_{rules} by *Eq.* 6.8.
  foreach $p \in \mathscr{P}$ **do**
    Calculate $\texttt{SRM}(p, p')$ for each $p' \in \mathscr{P} - \{p\}$.
    Update L_{clus}.
  $L = L_{\text{matched}} + \alpha \cdot L_{\text{rules}} + \beta \cdot L_{\text{clus}} + \gamma \cdot L_{\text{unmatched}}$.
  Update model parameters *w.r.t. L*.

calculate the average cross-entropy loss in the batch by L_{matched}. Similarly, we sample $\mathscr{B}_{\text{u}}$ from $\mathscr{S}_{\text{unmatched}}$, and generate the pseudo-labels with the entire rule set $\mathscr{P}$, and calculate normalized weights for $L_{\text{unmatched}}$ by:

$$
\begin{aligned}
w_s &= \frac{\exp(\sigma \texttt{SRM}(s, \hat{p}_i))}{\sum_{s' \in \mathscr{B}_{\text{u}}} \exp(\sigma \texttt{SRM}(s', \hat{p}_j))}, \\
L_{\text{unmatched}}(\theta_{RC}) &= \frac{1}{|\mathscr{B}_{\text{u}}|} \sum_{s \in \mathscr{B}_{\text{u}}} \left[-w_s \log \mathbb{P}_{\theta_{RC}}(r = r_{\hat{p}} | s) \right].
\end{aligned}
\tag{6.12}
$$

The normalized instance weights ensure the scale of $L_{\text{unmatched}}$ to be stable across different steps. Finally, we calculate L_{rules} and L_{clus} with the entire $\mathscr{P}$. For L_{rules}, the loss is averaged for all $p \in \mathscr{P}$. For L_{clus}, we take all $p \in \mathscr{P}$ to calculate the contrastive loss w.r.t. $\mathscr{P} - \{p\}$, and average the losses. We do not sample $\mathscr{P}$ due to the relatively small number of rules and simple structure of the soft rule matcher (SRM). We stop training when the joint training loss L converges and take RC as the output. Note that word vectors are also trainable parameters that are initialized by pre-trained embeddings.

6.4.2 Model Inference

Model inference in NERO aims at predicting the relation of a new sentence. Intuitively, we can give out a prediction by passing the sentence through the RC and return the relation with the highest probability as the prediction. As an alternative, we can also predict the relation using the SRM. Given a sentence, we can first find the most similar rule and return its rule head as the prediction. This alternative way of inference can be applied for predicting *unseen relations* given new rules in the

Dataset	# Train / Dev / Test	# Relations	# Rules	# matched Sent.
TACRED [20]	75,049 /25,763 / 18,659	42	270	1,630
SemEval [230]	7,199 /800 / 1,864	19	164	1,454

Table 6.1: Statistics for TACRED and SemEval datasets.

testing time. In experiments, the first method shows much better performance, since RC can capture rich contextual information while the SRM cannot.

For predicting "NONE" examples, our model filters out the predictions that our model is least certain[4] about. Specifically, we measure the uncertainty using the entropy of softmax distribution or the similarity score produced by SRM.

6.5 Experiments

In this section, we first introduce the datasets and compared baseline methods. Then, we illustrate the detailed setup and present extensive experiments with discussion and analysis. Finally, we conduct user studies to investigate the efficiency of our proposed approach.

6.5.1 Data Preparation

We choose two public and widely-used sentence-level relation extraction datasets in our experiments (see Tab. 6.1) as follows. For both datasets, we construct the rule set as illustrated in Section 6.3.2[5].

- **TACRED** [20] (TAC relation extraction dataset) contains more than 100,000 sentences categorized into 42 relation types. Among the sentences, 79.5% of the examples are labeled as NONE. We construct 270 rules which 1,630 hard-matched sentences in the official training data;
- **SemEval** 2010 Task 8 [230] contains about 10,000 sentences with 19 relation types, where 17.4% of the sentences are NONE. We construct 164 rules which hard-match 1,454 instances in the official training data.

6.5.2 Compared Methods

Recall Sec. 6.3.5, we first apply hard-matching on the unlabeled sentences and partition them into hard-matched sentences ($\mathscr{S}_{\text{matched}}$) and unmatched sentences ($\mathscr{S}_{\text{unmatched}}$). Our framework is applicable in both training with $\mathscr{S}_{\text{matched}}$ and training with $\mathscr{S}_{\text{matched}} \cup \mathscr{S}_{\text{unmatched}}$. Thus, we compare our models with *rule-based methods*, *supervised methods* and *semi-supervised methods*. For both semi-supervised and supervised methods, we use hard-matched sentences as the training data for comparing under the same setting. The experiments evaluate the ability to use rules for improving RE.

[4] A threshold δ tuned on dev set is used.

[5] We filter candidate patterns by requiring frequency ≥ 3.

Rule-based Baseline Methods. We apply the following models on $\mathscr{S}_{\text{matched}} \cup \mathscr{S}_{\text{unmatched}}$: (1) `CBOW-GloVe` adopts continuous bag-of-words [242] on GloVe embeddings [243] to represent a sentence or rule body, which labels a sentence using its most similar rule (in cosine distance). (2) `BREDS` [227] is a rule-based bootstrapping method originally designed for corpus-level RE. Given some entity pairs as seeds, it alternates between extracting new rules and new entity pairs. BREDS performs soft matching between rules and sentences using average word embeddings. In our experiments, we apply the set of rules learned by BREDS to perform hard-matching over the test sentences at the prediction time. (3) The Neural Rule Engine (`NRE` [244]) is an unsupervised method of soft-matching. It generalizes given rules by first performing unigram matching for each token in the rules using CNN, then accumulates the matching scores of all tokens along the parsed tree structure. When used for prediction, NRE performs soft-matching on the given rules.

Supervised Baseline Methods. We apply the following methods on $\mathscr{S}_{\text{matched}}$: (1) `PCNN` [220] represents each token using both word embeddings and positional embeddings. Convolution and piecewise max pooling layer are performed to produce the sentence embedding. (2) `LSTM+ATT` adopts bi-directional LSTM and attention mechanism [237] to produce a sentence embedding, which is fed into a fully-connected layer and a softmax classifier to predict the relation. (3) `PA-LSTM` [20] extends the LSTM model by incorporating positional information into attention mechanism and achieved state-of-the-art performance on TACRED. (4) `Data Programming` [89, 228] denoises the conflicting rules by learning their accuracy and correlation structures. (5) `LSTM+ATT` $(\mathscr{S}_{\texttt{matched}} + \mathscr{P})$ extends the LSTM+ATT model by also using rules as training data. It serves as the base model for all semi-supervised baselines.

Semi-Supervised Baseline Methods. To make fair comparisons with NERO, we also regard labeling rules as training data (same to L_{rules}) and use `LSTM+ATT` $(\mathscr{S}_{\texttt{matched}} + \mathscr{P})$ as the base model. We apply the following methods on $\mathscr{S}_{\text{matched}} \cup \mathscr{S}_{\text{unmatched}} \cup \mathscr{P}$: (1) `Pseudo-Labeling` [240] trains the network using labeled and unlabeled data simultaneously. Pseudo-Labels are created for unlabeled data by picking up the class with the maximum predicted probability and are used as if they are labeled data during training. (2) `Self-Training` [229] iteratively trains the model using the labeled dataset and expands the labeled set using the most confident predictions among the unlabeled set. This procedure stops when unlabeled data is exhausted. (3) `Mean-Teacher` [245] assumes that data points with small differences should have similar outputs. We perturb each unlabeled sentence using word dropout and regularize their outputs to be similar. (4) `DualRE` [231] jointly trains a relation prediction and a retrieval module which mutually enhance each other by selecting high-quality instances from unlabeled data.

Variants of NERO. (1) `Nero w/o` $\mathscr{S}_{\texttt{unmatched}}$ removes the loss on unmatched sentences and only use $\mathscr{S}_{\text{matched}}$ and $\mathscr{P}$ in model training. It only performs hard-matching on rules. (2) `Nero-SRM` has the same training objective as NERO, but uses the soft rule matcher (`SRM`) to make predictions. Given a sentence, NERO-SRM finds the most similar rule and returns the rule head as the prediction. It can be taken as a context-agnostic version of NERO (Sec. 6.4.2).

6.5.3 Experiment Settings

Implementation. We implement most baselines from the scratch using Tensorflow 1.10 [246] except for those that have released their codes (like `PA-LSTM` and `DualRE`). We adapt the baselines

Method / Dataset	TACRED			SemEval		
	Precision	Recall	F_1	Precision	Recall	F_1
Rules	85.0	11.4	20.1	81.2	17.2	28.5
BREDS [227]	53.8	20.3	29.5	62.0	24.5	35.1
CBOW-GloVe	27.9	45.7	34.6	44.0	52.8	48.0
NRE [244]	65.2	17.2	27.2	78.6	18.5	30.0
PCNN [220]	44.5 ± 0.4	24.1 ± 2.8	31.1 ± 2.6	59.1 ± 1.4	43.0 ± 0.7	49.8 ± 0.5
LSTM+ATT	38.1 ± 2.7	39.6 ± 2.7	38.8 ± 2.4	64.5 ± 2.8	53.3 ± 2.8	58.2 ± 0.8
PA-LSTM [20]	39.8 ± 2.5	40.2 ± 2.0	39.0 ± 0.6	64.0 ± 3.6	54.2 ± 2.5	58.5 ± 0.6
Data Programming [89]	39.2 ± 1.3	40.1 ± 2.0	39.7 ± 0.9	61.8 ± 2.1	54.8 ± 1.1	58.1 ± 0.7
LSTM+ATT ($\mathscr{S}_{\text{matched}} + \mathscr{P}$)	39.2 ± 1.7	45.5 ± 1.7	42.1 ± 0.9	63.4 ± 2.1	55.0 ± 0.3	58.8 ± 0.9
Pseudo-Labeling [240]	34.5 ± 4.1	37.4 ± 5.1	35.3 ± 0.8	59.4 ± 3.3	55.8 ± 2.1	57.4 ± 1.3
Self-Training [229]	37.8 ± 3.5	41.1 ± 3.1	39.2 ± 2.1	62.3 ± 2.0	53.0 ± 2.7	57.1 ± 1.0
Mean-Teacher [245]	46.0 ± 2.7	41.6 ± 2.2	43.6 ± 1.3	62.3 ± 1.5	54.5 ± 1.2	57.9 ± 0.5
DualRE [231]	40.2 ± 1.5	42.8 ± 2.0	41.7 ± 0.5	63.7 ± 2.8	54.6 ± 2.1	58.6 ± 0.8
NERO w/o $\mathscr{S}_{\text{unmatched}}$	41.9 ± 1.8	44.3 ± 3.8	42.9 ± 1.4	61.4 ± 2.4	56.2 ± 1.9	58.6 ± 0.6
NERO-SRM	45.6 ± 2.2	45.2 ± 1.2	45.3 ± 1.0	54.8 ± 1.6	55.2 ± 2.0	54.9 ± 0.6
NERO	54.0 ± 1.8	48.9 ± 2.2	$\mathbf{51.3 \pm 0.6}$	66.0 ± 1.5	55.8 ± 0.9	$\mathbf{60.5 \pm 0.7}$

Table 6.2: Performance comparison (in %) of relation extraction on the TACRED and SemEval datasets. We report the mean and standard deviation of the evaluation metrics by conducting 5 runs of training and testing using different random seeds. We use LSTM+ATT ($\mathscr{S}_{\text{matched}} + \mathscr{P}$) as the base model for all semi-supervised baselines and our models.

to our setting. For supervised and semi-supervised baselines, we use the hard-matched sentences ($\mathscr{S}_{\text{matched}}$) as the "labeled data" and the unmatched sentences ($\mathscr{S}_{\text{unmatched}}$) as the "unlabeled data".
Training details. We use pre-trained Glove embeddings [243] to initialize the word embeddings and fine-tune them during training. For NERO, We set the batch size to 50 for hard-matched sentences, and 100 for unmatched sentences. For other baselines, we set the batch size to 50. To exclude the possibility that NERO takes advantage of a larger batch size, we also tried a batch size of 150 for other baselines but it showed no difference in performance. To reduce over-fitting, we adopt the entity masking technique [20] and replace the subject and object with SUBJ/OBJ–NER and regard them as normal tokens. We use a two-layer bi-directional LSTM as the encoder. The hidden dimension is 100 for the LSTM, and 200 for the attention layers. We set β, γ, τ to 0.05, 0.5, and 1.0 respectively. We set α to 1.0 for TACRED, and 0.1 for SemEval. The temperature θ in instance weighting is set to 10. We apply dropout [63] after the LSTM with a rate of 0.5. All models are optimized using AdaGrad [247] with initial learning rate 0.5 and decay rate 0.95. For models that require the prediction of NONE (including NERO and all supervised / semi-supervised baselines), we select the threshold (see 6.4.2) from range [0.0, 1.0] based on the dev set.

6.5.4 Performance Comparison

Rule-based Models. We first compare the models that solely use rules, as shown in Table 6.2. Hard-matching with rules achieves a high precision (85% on TACRED) but suffers from the low-recall problem due to its failure in matching instances with similar meanings (only 11.4% on TACRED). Bootstrapping methods (such as BREDS) and unsupervised soft-matching methods (such as NRE)

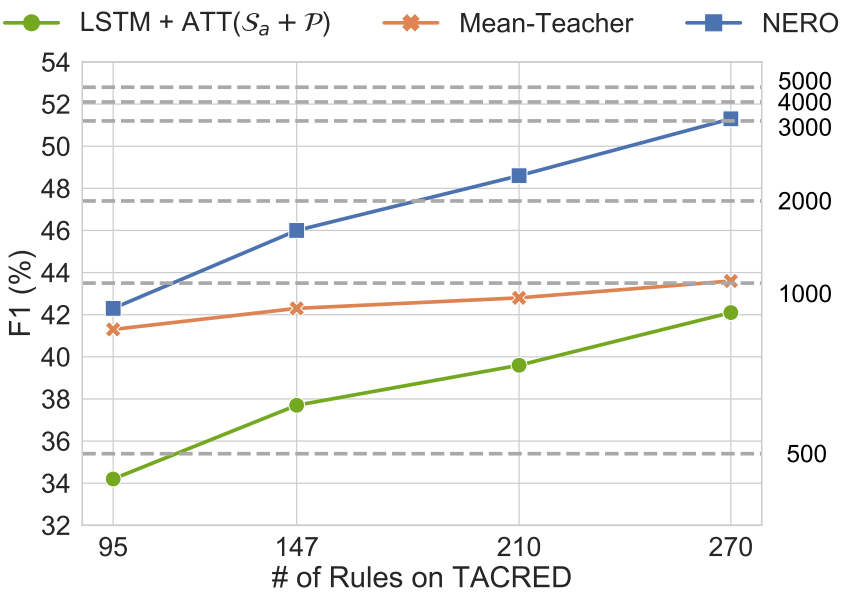

Figure 6.6: Performance w.r.t. different number of rules and human-annotated labels on TACRED. We show different models' F1 scores and number of rules or labels used for training the corresponding model.

manage to cover more instances (6% gain in recall on TACRED dataset), but they fail to capture contextual information.

Models Trained on $\mathcal{S}_{\text{matched}}$. Neural networks are capable of fitting the hard-matched data, but they suffer from over-fitting due to the small data size. Adding the pattern encoder loss (L_{rules}) and the soft matching loss (L_{clus}) boosts the performance by a huge margin, as indicated by LSTM+ATT ($\mathcal{S}_{\text{matched}} + \mathcal{P}$) and NERO w/o $\mathcal{S}_{\text{unmatched}}$). However, their performance is still far from satisfactory, since they cannot utilize large amounts of unmatched sentences. Data programming does not bring any improvement because our rule mining and labeling method rarely introduces conflicted labeling rules. Hence, data programming with our rules is the same as LSTM+ATT.

Models Trained on $\mathcal{S}_{\text{matched}} \cup \mathcal{S}_{\text{unmatched}} \cup \mathcal{P}$. Table 6.2 shows the results for methods that incorporate unmatched data and rules. For methods that actively create labels for unmatched data, namely pseudo-labeling, self-training and DualRE, their performance is even worse than the supervised counterparts (LSTM+ATT ($\mathcal{S}_{\text{matched}} + \mathcal{P}$)). It is because in a low-resource setting, the model performance is so low that the created labels are very noisy. Mean-Teacher manages to improve model performance on TACRED, but only by a small margin. Our proposed model, NERO, is relatively indifferent to the noise due to the soft rule matcher, which learns directly from similar textual sentences. This results in a significant gain in precision (14% on TACRED) and recall (4% on TACRED) over the semi-supervised baselines. Compared to TACRED, the improvement on SemEval is much smaller (only 1.7% in F1). We hypothesize it is because the sentences in SemEval are quite short and contain very simple rule patterns. Thus, the soft-matched sentences can hardly provide additional information. We also try to predict with the soft rule matcher, but the performance is lower in F1 (45.3% on TACRED and 54.9% on SemEval) since it cannot capture contextual information.

6.5.5 Performance Analysis

1. Performance on Various Amounts of Rules and Human Annotated Labels. To show that our rules are more powerful than human-annotated labels, we show the performance of models trained with different number of rules (30%, 50%, 70%, and 100%) and the number of labels required to reach comparable performance using supervised baseline (LSTM+ATT). We random sample subsets of rules w.r.t. each relation to avoid extreme cases where some relations are not sampled at all. We conduct 5 runs of sampling and training and report the average performance. As shown in Figure 6.6, NERO consistently outperforms other methods by a huge margin, and its performance increases with the number of rules (from 42.3% to 51.3%). Mean-Teacher also outperforms the base model especially when the number of rules is small, but its performance does not increase much (from 41.3% to 43.6%). It shows that Mean-Teacher generalizes well in low-resource setting but cannot fully utilize the knowledge in rules. In terms of label efficiency, one rule in NERO is equivalent to 10 human-annotated labels in model performance. Even in the base model LSTM+ATT ($\mathscr{S}_{\text{matched}} + \mathscr{P}$), one rule is still equivalent to about 4 labels. It demonstrates the superior generalization ability of rules over labels.

2. Performance on Various Amounts of the Raw Corpus. To test the robustness of our model, we plot the performance curve in Fig. 6.7 when different amounts of raw corpus are available on TACRED. Again, NERO outperforms all methods, and its F_1 score positively correlates to the amount of available data (from 43.4% to 51.3%). We also observe similar phenomena in other methods (all based on LSTM+ATT ($\mathscr{S}_{\text{matched}} + \mathscr{P}$) model), which show that matching on rules provides additional knowledge to model training. Just memorizing these rules with a neural model leads to very bad performance. Also, we observe that with only 10% of data, our final model (NERO) already outperforms the best supervised baseline NERO w/o $\mathscr{S}_{\text{unmatched}}$. This indicates that the soft matching module can utilize the raw corpus and rules in a more data-efficient way.

3. Performance on Unseen Relations. To show that NERO has the capacity of predicting unseen relations, we evaluate NERO on relations unseen in training. In the experiment, we randomly sample 5 relations as unseen relations and repeat the experiment 10 times. In each time, we use the same sets of unseen relations and the same random seed for all methods. We remove the sampled relations (both sentences and rules) in training while keeping only them in testing. We sample sentences of other relations as NONE relation in testing with the same ratio as the raw corpus (79.5% for TACRED and 17.4% for SemEval). Moreover, since the special tokens SUBJ/OBJ-NER are out-of-vocabulary in both Glove and BERT, we only use the words between two entities in encoding and only use rules with the same entity types to the sentence in inference. To make predictions by NERO, we find the most similar rule using soft rule matcher. Baseline models include exact rule matching, cosine similarity of average Glove embedding, and cosine similarity of pre-trained BERT embedding [14]. As shown in Table 6.3, our soft rule matcher achieves similar or better performance compared to all baselines. It shows that instead of just memorizing the rules and the corresponding relation types, the soft rule matcher also learns knowledge about how to better perform rule matching and can even generalize to unseen relations.

6.5.6 Model Ablation Study

1. Effect of Different Loss Functions. To show the efficacy of our proposed losses (L_{clus}, $L_{\text{unmatched}}$ and L_{rules}), we conduct ablation study on loss functions by removing one loss function at a time. As

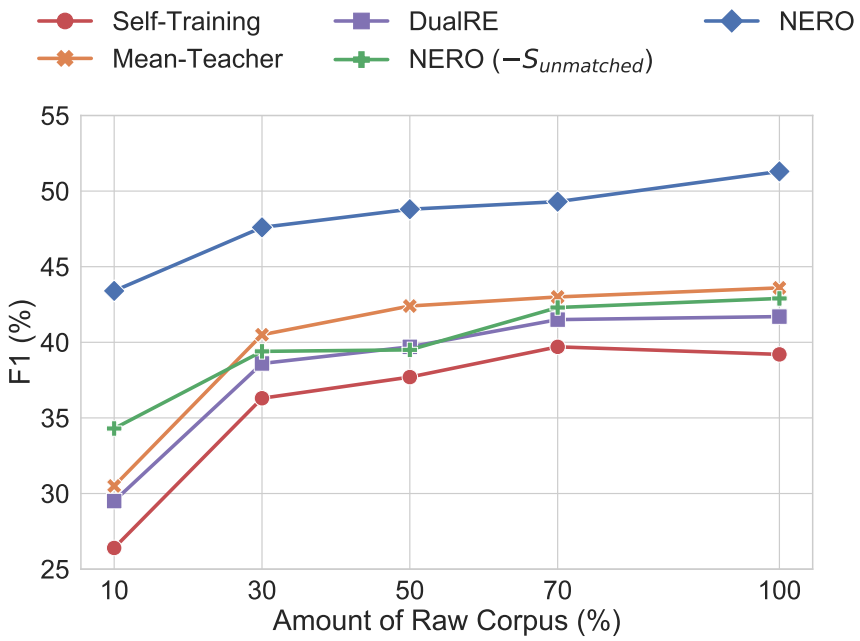

Figure 6.7: Performance of different semi-supervised models trained using various amounts of unlabeled sentences randomly sampled from the raw corpus on TACRED.

	TACRED			SemEval		
Method	P	R	F_1	P	R	F_1
Rule (exact match)	100	6.1	10.8	83.2	17.7	28.2
CBOW-GloVe	52.4	86.3	64.7	40.3	45.5	34.7
BERT-base (frozen)	66.2	76.8	**69.5**	37.8	33.2	35.3
NERO	61.4	80.5	68.9	43.0	54.1	**45.5**

Table 6.3: Performance on predicting unseen relations. NERO applies the learned soft rule matcher on unseen relation rules to make predictions.

shown in Table 6.4, all three losses contribute to the final performance, while $L_{\text{unmatched}}$ helps the most, which proves the effectiveness of soft-matching. When removing the contrastive loss L_{clus}, the F1 score drops from 51.3 to 46.4, which shows the effectiveness of a trainable SRM. The L_{rules} loss also brings improvement, similar to including more training instances.

Objective	Precision	Recall	F_1
L (ours)	54.0	48.9	51.3
$-L_{\text{rules}}$	50.0	47.7	49.0
$-L_{\text{clus}}$	50.9	43.0	46.4
$-L_{\text{unmatched}}$	41.9	44.3	42.9

Table 6.4: Ablation Study of Different Training Objectives on TACRED dataset. We remove each loss term one at a time.

2. Effect of Different Soft-matching Models. Besides word-level attention, we also try other soft matching mechanisms including CBOW-Glove, LSTM+ATT, and BERT-base (both frozen and fine-tuned). All methods first summarize the sentences / rules into vector representations, then calculate the matching scores by cosine similarity. We report their performance in Table 6.5. We observe that our method achieves the best performance despite its simplicity. To our surprise, BERT-base performs much worse than our method. In the experiment we find that BERT tends to given high matching scores for all sentence-rule pairs so our framework cannot distinguish the false pseudo-labels from reliable ones.

Objective	Precision	Recall	F_1
CBOW-Glove	49.4	43.5	46.2
LSTM+ATT	56.2	46.0	50.6
BERT-base (frozen)	45.6	47.6	46.5
BERT-base (fine-tuned)	50.3	45.8	47.9
Word-level attention (ours)	54.0	48.9	51.3

Table 6.5: Ablation study of different soft-matching models for NERO on the TACRED dataset.

3. Sensitivity Analysis of Model Hyper-parameters. The most important hyper-parameters in NERO are α, β, γ, and τ. We test the sensitivity of these parameters on TACRED. We adjust one hyper-parameter at a time and remain the other three unchanged. The results are shown in Figure 6.8. We observe that for τ and β, there exists a wide interval where the F1 score remains stable. For α and γ, the F1 score hardly changes with the choice of their values. This is because that L_{matched}, L_{clus}, and L_{rules} quickly drop to 0 and $L_{\text{unmatched}}$ dominates the joint loss in the training phase, and the impact of γ is lessened by adaptive subgradient method (AdaGrad).

6.5.7 Case Study

Study on Label Efficiency. To test the label efficiency of rules and sentences in real scenarios, we ask 5 college students in computer science to annotate frequent candidate rules (with frequency ≥ 3) and the unlabeled sentences, during a 40-minute period. For annotating sentences, they are required to provide labels from the pre-defined relation set $R \cup \text{NONE}$. For annotating candidate rules, they are required to filter out uninformative rules (e.g. "SUBJ-PERSON and OBJ-PERSON") and assign labels to the remaining ones. For each candidate rule, they are further given an example of hard-matched sentence to help understand the meaning and context of the rule. Then we use each user's annotations to train the RE model. We use **LSTM+ATT** for labeled sentences and **NERO** for rules.

We report the average number of selected rules / labeled sentences and performance of trained RE model every 5 minutes in Figure 6.10. We observe that getting rules is nearly as fast as getting labels but is much more efficient. With the same annotation time, NERO can already learn a good RE classifier (with 41.3% F1), while LSTM+ATT cannot learn anything from the labels. In just 40 minutes, NERO can achieve comparable performance to 1000 labels (refer to Figure 6.6), which requires about 320 minutes to annotate.

Interpretability of Soft Rule Grounding. Fig. 6.9 gives an example of the similarity matrix, showing that our soft matching module learns meaningful representations of rules for soft matching.

The left part of the figure presents 1) which words are more important for building attention-pooled rule/sentence representations, and 2) the soft matching scores between the rule and three sentences. We also show two concrete examples of attention matrix in the right part. For each word-pair between a rule and a sentence, we visualize the cosine similarity matrix between word representations learned with the contrastive loss. While the word "chairman" in our rule matches perfectly with the same surface word in the sentence, it also has a high similarity score with "Chief" and "Executive".

6.6 Related Work

Relation Extraction. Traditional corpus-level RE systems take a rule-based approach, in which rules are either handcrafted [248] or automatically learned from large corpora [226, 227]. Although feature-based soft matching methods (e.g. TF-IDF, CBOW) are used in rule mining, the inference (i.e, labeling new sentences) is still done by hard-matching, which leads to high precision but low recall due to the low coverage of rules. While we use a trainable soft rule matcher, which is more powerful than prior feature-based methods. And we apply soft-matching to both training and inference phase in a consistent manner. Recent RE models [20, 220, 249] successfully apply deep neural networks with the help of large scale datasets such as SemEval [230] and TACRED [20]. However, their performance degrades heavily when labeled data is insufficient. Some work [228, 250] proposes to train a RE classifier from natural language explanations, which are first transformed to labeling rules by a semantic parser then applied to raw corpora by hard-matching. But none of them utilizes the data that cannot be matched by rules, which is of massive amount.

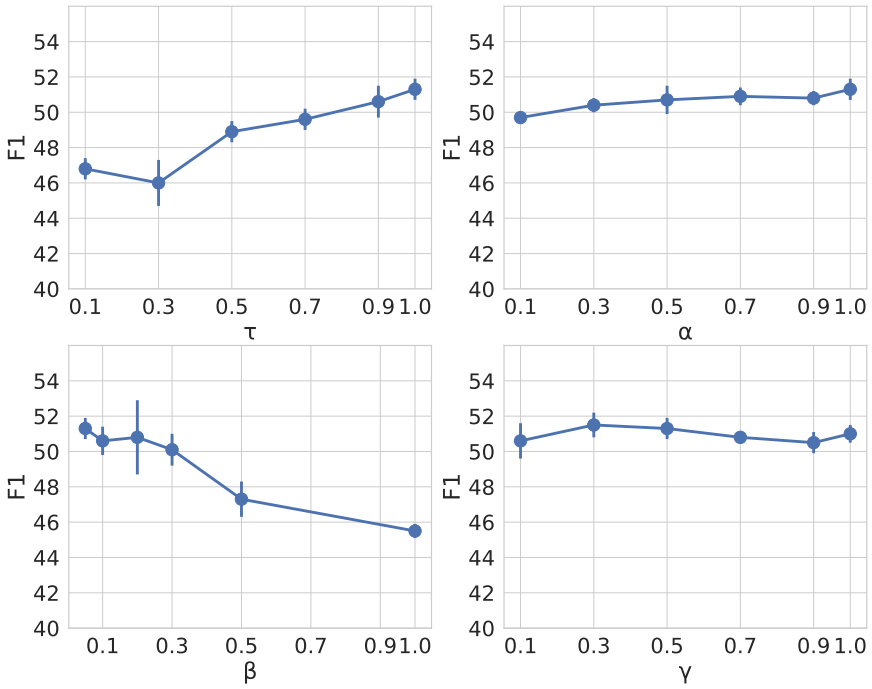

Figure 6.8: Sensitivity analysis of τ, α, β, and γ on TACRED. We report the mean and standard deviation F1 score by conducting 5 runs of experiments using different random seeds.

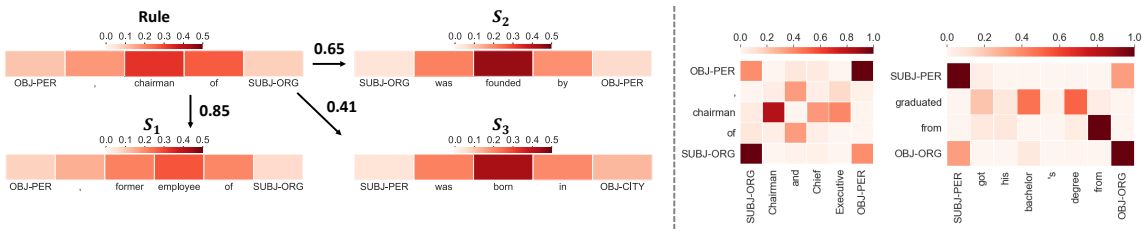

Figure 6.9: Output visualization of SRM. Left: attention weights of words and the soft matching scores between a rule and three sentences. Right: cosine similarity matrix between word embeddings learned with the contrastive loss.

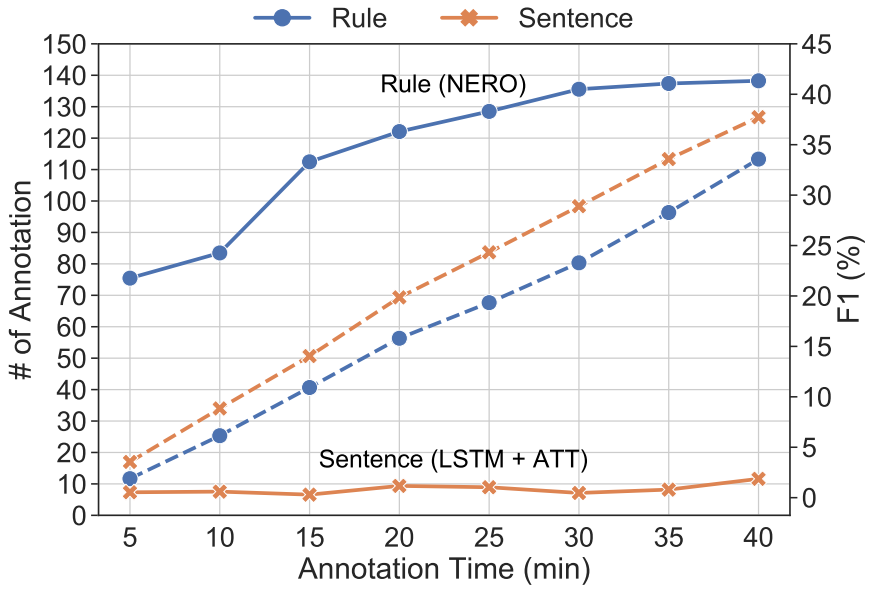

Figure 6.10: Study on label efficiency. Average number of rules / sentences labeled by annotators (dashed line) are shown on the x-axis over the left-hand side; and the performance of models trained with these corresponding labeled rules / sentences (solid line) are shown on the x-axis over the right-hand side. We use NERO and LSTM+ATT as the base model for the labeling rules and sentences, respectively.

This motivates us to propose a framework that can fully utilize the information from both rules and raw corpora.

Semi-Supervised Learning. Our work is relevant to semi-supervised learning, if we consider rule-annotated data as labeled among the corpus, e.g., self-training [229], mean-teacher [245], and semi-supervised VAE [251]. In relation extraction, one line of work proposes to alleviate the low recall problem in rule-based approach using bootstrapping framework [226, 252] and revised pattern matching [227, 253]. However, their settings are tailored for corpus-level RE. Another line of work applies self-training framework in supervised learning models. However, These models turn out to be ineffective in rule-labeled data due to potentially large difference in label distribution, and the generated psuedo-labels may be quite noisy.

6.7 Conclusion

We proposed a novel framework, named NERO, for label-efficient relation extraction. We automatically extracted frequent patterns from large raw corpora and asked users to annotate them to form labeling rules. To increase the coverage of these rules, we further proposed the soft-matching mechanism, where unlabeled sentences are annotated by their most semantically similar labeling rules and weighted in training the RE model. Experiments on two public datasets proved the effectiveness of our framework.

Chapter 7

Continual Contrastive Finetuning

Relation extraction (RE), which has relied on structurally annotated corpora for model training, has been particularly challenging in low-resource scenarios and domains. Recent literature has tackled low-resource RE by self-supervised learning, where the solution involves pretraining the relation embedding by RE-based objective and finetuning on labeled data by classification-based objective. However, a critical challenge to this approach is the gap in objectives, which prevents the RE model from fully utilizing the knowledge in pretrained representations. We aim at bridging the gap and propose to pretrain and finetune the RE model using consistent objectives of contrastive learning. Since in this kind of representation learning paradigm, one relation may easily form multiple clusters in the representation space, we further propose a multi-center contrastive loss that allows one relation to form multiple clusters to better align with pretraining. Experiments on two document-level RE datasets, BioRED and Re-DocRED, demonstrate the effectiveness of our method. Particularly, when using 1% end-task training data, our method outperforms PLM-based RE classifier by 10.5% and 6.1% on the two datasets, respectively[1].

7.1 Introduction

Relation extraction (RE) is a fundamental task in NLP. It aims to identify the relations among entities in a given text from a predefined set of relations. While much effort has been devoted to RE in supervised settings [20, 30, 255], RE is extremely challenging in high-stakes domains such as biology and medicine, where annotated data are comparatively scarce due to overly high annotation costs. Therefore, there is a practical and urgent need for developing low-resource RE models without the reliance on large-scale end-task annotations.

To realize low-resource RE, previous work has focused on pretraining relation mention representations on large corpora using RE-based pretraining objectives. Particularly, Baldini Soares, FitzGerald, Ling & Kwiatkowski [256] propose a self-supervised matching-the-blanks (MTB) objective that encourages the representations of pairs of relation mentions to be similar if they contain the same entity pairs. Later work [257, 258] extends this idea by generating relation mention pairs with distant supervision [259] and improves representation learning using contrastive learning [136, 137, 260]. To adapt to training on RE annotations, these works finetune pretrained relation mention representations on labeled data using classification-based objectives. Although this paradigm produces better results compared to RE models initialized with pretrained language

[1]This chapter is based on Zhou, Zhang, Naumann, Chen & Poon [254].

models (PLMs), it creates a significant divergence between pretraining and finetuning objectives, thus preventing the model from fully exploiting knowledge in pretrained representations.

We aim to bridge this gap in RE pretraining and finetuning. Our key idea is to use similar objectives in pretraining and finetuning. First, we propose to continually finetune the representations by contrastive learning, which encourages the representations of relation mentions corresponding to the same relation to be similar. However, as pretraining and finetuning are conducted on different tasks, relation mentions of the same relation can form multiple different clusters in the pretrained representation, where standard supervised contrastive loss [154] may distort the representation because of its underlying one-cluster assumption [261]. Therefore, we further propose a multi-center contrastive loss (MCCL), which encourages a relation mention to be similar to only a subset of mentions of the same relation type, allowing one relation to have multiple clusters. Second, we propose to use k-nearest neighbors (kNN; [262, 263]) in inference, where predictions are made based on most similar instances.

We focus our work on document-level RE [21, 264], which consists of both intra- and cross-sentence relations. To the best of our knowledge, this work represents the first effort to explore self-supervised pretraining for document-level RE. Unlike prior studies [257, 258], we do not use distant supervision. Instead, we pretrain relation mention representations with an improved MTB objective on unlabeled corpora, where we use contrastive learning to learn representations that suit downstream RE. We then finetune the pretrained model on labeled data with MCCL. Experiments on two datasets, BioRED [11] in the biomedical domain and Re-DocRED [265] in the general domain, demonstrate that our pretraining and finetuning objectives significantly outperform baseline methods in low-resource settings. Particularly, in the low-resource setting of using 1% of labeled data, our method outperforms PLM-based classifiers by 10.5% and 6.1% on BioRED and Re-DocRED, respectively. Based on our pretrained representations, MCCL outperforms classification-based finetuning by 6.0% and 4.1%, respectively.

Our technical contributions are three-fold. First, we propose to pretrain the PLMs based on our improved MTB objective and show that it significantly improves PLM performance in low-resource document-level RE. Second, we present a technique that bridges the gap of learning objectives between RE pretraining and finetuning with continual contrastive finetuning and kNN-based inference, helping the RE model leverage pretraining knowledge. Third, we design a novel MCCL finetuning objective, allowing one relation to form multiple different clusters, thus further reducing the distributional gap between pretraining and finetuning.

7.2 Related Work

Document-level RE. Existing document-level RE models can be classified into graph-based and sequence-based models. Graph-based models construct document graphs spanning across sentence boundaries and use graph encoders such as the graph convolution network (GCN; [66]) to aggregate information. Particularly, Quirk & Poon [266] build document graphs using words as nodes with inner- and inter-sentence dependencies (e.g., syntactic dependencies, coreference, etc.) as edges. Later work extends this idea by applying different network structures [264, 267] or introducing other node types and edges [255, 268, 269]. On the other hand, sequence-based methods [27, 270, 271] use PLMs to learn cross-sentence dependencies without using graph structures. Particularly, Zhou, Huang, Ma & Huang [27] propose to enrich relation mention representation by localized context

pooling. Zhang, Chen, Xie, Deng, Tan, Chen, *et al.* [270] propose to model the inter-dependencies between relation mentions by semantic segmentation [272]. In this work, we study a general method of self-supervised RE. Therefore, our method is independent of the model architecture and can be adapted to different RE models.

Low-resource RE. Labeled RE data may be scarce in real-world applications, especially in low-resource and high-stakes domains such as finance and biomedicine. Much effort has been devoted to training RE models in low-resource settings. Some work tackles low-resource RE by indirect supervision, which solves RE by other tasks such as machine reading comprehension [273], textual entailment [25], and abstractive summarization [274]. However, indirect supervision may not be practical in high-stake domains, where annotated data for other tasks are also scarce. Other efforts [256–258, 275, 276] improve low-resource RE by pretraining on large corpora. Specifically, Baldini Soares, FitzGerald, Ling & Kwiatkowski [256] propose an MTB objective that encourages the representations of relation mentions containing the same entity pairs to be similar. Peng, Gao, Han, Lin, Li, Liu, *et al.* [257] propose to pretrain on distantly labeled corpora, where they seek to represent the relation mentions of the same distant label to be similar. They also introduce a contrastive learning based training objective to improve representation learning. Qin, Lin, Takanobu, Liu, Li, Ji, *et al.* [258] further introduce an entity discrimination task and pretrain the RE model on distantly labeled document corpora. We study self-supervised pretraining for document-level RE. We study how to reduce the gap between pretraining and finetuning, which is critical to bridge the training signals obtained in these two stages but has been overlooked in prior work.

7.3 Method

In this work, we study a self-supervised approach for document-level RE. Given a document d and a set of entities $\{e_i\}_{i=1}^{N}$, where each entity e_i can have one or multiple entity mentions in the document, document-level RE aims at predicting the relations of all entity pairs $(e_s, e_o)_{s,o \in \{1,\ldots,N\}}$ from a predefined set of relationships $\mathscr{R}$ (including an NA class indicating no relation exists), where e_s and e_o are the subject and object entities, respectively. In the self-supervised RE setting, we have a large unlabeled document corpus for pretraining and a labeled RE dataset for finetuning. The document corpus has been annotated with entity mentions and the associated named entity types but no relations. Our goal is to train a document-level RE classifier, especially in the low-resource setting.

Our training pipeline consists of two phases: pretraining and finetuning. In pretraining, we use the (unlabeled) document corpus to pretrain the relation mention representations based on our improved *matching-the-blanks* training objective (MTB; [256]), where the LM learns to decide whether two relation mentions contain the same entity pairs or not, and the learning of representation is enhanced with contrastive learning. In finetuning, we continue to train the pretrained model on relation-labeled data using a multi-center contrastive loss (MCCL), which achieves better performance than the traditional classifier paradigm due to the better-aligned learning objective with pretraining. After training, we use k-nearest neighbor (kNN) inference that suits well the contrastively finetuned model.

The rest of this section is organized as follows: we introduce the model architecture used in both pretraining and finetuning in Section 7.3.1, the pretraining process in Section 7.3.2, finetuning in Section 7.3.3, and inference in Section 7.3.4.

7.3.1 Model Architecture

Encoder. Given a document $d = [x_1, x_2, ..., x_l]$, we first mark the spans of the entity mentions by adding special entity markers [E] and [/E] to the start and the end of each mention. Then we encode the document with a PLM to get the contextual embedding of textual tokens:

$$\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_l] = \text{PLM}([x_1, x_2, ..., x_l]).$$

We take the contextual embedding of [E] at the last layer of the PLM as the embedding of entity mentions. We accumulate the embedding of mentions corresponding to the same entity by LogSumExp pooling [264] to get the entity embedding $\boldsymbol{h}_{e_i}$.

Relation Embedding. Given a relation mention $t = (e_s, e_o, d)$, where e_s and e_o are the subject and object entities, respectively, we represent the relation mention as:

$$\boldsymbol{z}^t = \boldsymbol{W}_{\text{linear}} \left[\boldsymbol{h}_{e_s}, \boldsymbol{h}_{e_o}, \boldsymbol{c}^{(e_s, e_o)} \right].$$

Here $\boldsymbol{h}_{e_s}, \boldsymbol{h}_{e_o} \in \mathbb{R}^d$ are the entity embedding vectors, $\boldsymbol{c}_{e_s, e_o} \in \mathbb{R}^d$ is the localized context encoding for (e_s, e_o), $\boldsymbol{W}_{\text{linear}} \in \mathbb{R}^{3d \times d}$ is a linear projector. The localized context encoding is introduced by Zhou, Huang, Ma & Huang [27] to derive the context embedding conditioned on a relation mention, which finds the context that both the subject and object entities attend to. Specifically, denote the multi-head attention in the last layer of PLM as $\boldsymbol{A} \in \mathbb{R}^{m \times l \times l}$, where m is the number of attention heads, l is the input length, we first take the attention scores from [E] as the attention from each entity mention, then accumulate the attention of this entity mention by mean pooling to get the entity-level attention $\boldsymbol{A}^{(e_i)} \in \mathbb{R}^{m \times l}$. Finally, we compute $\boldsymbol{c}^{(e_s, e_o)}$ by:

$$\begin{aligned}
\boldsymbol{A}^{(e_s, e_o)} &= \boldsymbol{A}^{(e_s)} \odot \boldsymbol{A}^{(e_o)}, \\
\boldsymbol{q}^{(e_s, e_o)} &= \sum_{i=1}^{m} \boldsymbol{A}_i^{(e_s, e_o)}, \\
\boldsymbol{a}^{(e_s, e_o)} &= \boldsymbol{q}^{(e_s, e_o)} / \boldsymbol{1}^\top \boldsymbol{q}^{(e_s, e_o)}, \\
\boldsymbol{c}^{(e_s, e_o)} &= \boldsymbol{H}^\top \boldsymbol{a}^{(e_s, e_o)}.
\end{aligned}$$

We introduce in the rest of the section how to pretrain and finetune the RE model based on the relation embedding $\boldsymbol{z}^{(e_s, e_o)}$.

7.3.2 Pretraining

We pretrain the LM on the document corpus using the MTB objective. MTB is based on a simple assumption that, in contrast to different entity pairs, it is more often for the same entity pair to be connected with the same relation. The MTB objective transforms the similarity learning problem into a pairwise binary classification problem: given two relation-describing utterances where entity mentions are masked, the model classifies whether the entity pairs are the same or not. This pretraining objective has shown effectiveness in several sentence-level RE datasets[19, 20, 24].

However, when it comes to document-level RE, Qin, Lin, Takanobu, Liu, Li, Ji, *et al.* [258] have observed no improvement led by the vanilla MTB pretraining. Therefore, we replace the

pairwise binary classification with contrastive learning, which is adopted in later RE pretraining works [257, 258] and can effectively learn from more positive and negative examples. Details of training objectives are elaborated in the rest of the section.

Training objective. The overall goal of pretraining is to make the relation embedding of positive pairs closer than that of negative pairs. We use the InfoNCE loss [136] to model this objective. Given the documents in batch, $\mathscr{P}$ as the set of all positive relation mention pairs, and $\mathscr{N}_t$ denote the set of negative relation mentions w.r.t. relation mention t, the loss for relation mention pairs is[2]:

$$\mathscr{L}_{\text{rel}} = -\frac{1}{|\mathscr{P}|} \sum_{t_i, t_j \in \mathscr{P}} \log \frac{e^{\text{sim}(\boldsymbol{z}^{t_i}, \boldsymbol{z}^{t_j})/\tau}}{\mathscr{Z}_{t_i}}, \tag{7.1}$$
$$\mathscr{Z}_{t_i} = e^{\text{sim}(\boldsymbol{z}^{t_i}, \boldsymbol{z}^{t_j})/\tau} + \sum_{t_k \in \mathscr{N}_{t_i}} e^{\text{sim}(\boldsymbol{z}^{t_i}, \boldsymbol{z}^{t_k})/\tau},$$

where $\text{sim}(\boldsymbol{z}^{t_i}, \boldsymbol{z}^{t_j})$ denotes the similarity between the relation mention t_i and t_j. Following Chen, Kornblith, Norouzi & Hinton [137], we use cosine similarity as the similarity metric. Similar to SimCSE [277], we further add a self-supervised contrastive loss that requires the same relation embedding augmented by different dropout masks to be similar, thus encouraging the model to learn more instance discriminative features that lead to less collapsed representations. Specifically, denote the two relation embedding of t derived by different dropout masks as $\boldsymbol{z}^t$ and $\hat{\boldsymbol{z}}^t$, respectively, the set of all relation mentions in the batch as $\mathscr{T}$, and the set of relation mentions in positive pairs as $\mathscr{T}_P$, the self-supervised loss is formulated as:

$$\mathscr{L}_{\text{self}} = -\frac{1}{|\mathscr{T}_P|} \sum_{t_i \in \mathscr{T}_P} \log \frac{e^{\text{sim}(\boldsymbol{z}^{t_i}, \hat{\boldsymbol{z}}^{t_i})/\tau}}{\mathscr{Z}_{t_i}},$$
$$\mathscr{Z}_{t_i} = e^{\text{sim}(\boldsymbol{z}^{t_i}, \hat{\boldsymbol{z}}^{t_i})/\tau} + \sum_{t_k \in \mathscr{T} \setminus \{t_i\}} e^{\text{sim}(\boldsymbol{z}^{t_i}, \hat{\boldsymbol{z}}^{t_k})/\tau}.$$

Finally, we use a masked language model loss $\mathscr{L}_{\text{mlm}}$ to adapt the LM to the document corpus. The overall pretraining objective is:

$$\mathscr{L}_{\text{pretrain}} = \mathscr{L}_{\text{rel}} + \mathscr{L}_{\text{self}} + \mathscr{L}_{\text{mlm}}.$$

For faster convergence, we initialize our model with a PLM that is pretrained on a larger corpus, and continually pretrain the PLM on the document corpus with our new pretraining objectives. We use BERT [14] for the general domain and PubmedBERT [278] for the biomedical domain.

7.3.3 Finetuning

After pretraining, we finetune the LM on labeled document-level RE datasets. In previous studies [256–258], pretraining and finetuning are conducted in processes with different learning objectives. Specifically, after using the pretrained weights to initialize a RE classifier, the model is finetuned with a classification-based training objective. Based on our model architecture, a

[2] Similar to Baldini Soares, FitzGerald, Ling & Kwiatkowski [256], we randomly mask the entities in documents with a probability of 0.7 to avoid shortcut learning.

Classifier	BioRED	Re-DocRED
One-cluster		
Softmax	28.6	39.3
Nearest centroid	12.5	4.1
Multi-cluster		
kNN	**36.7**	**54.1**

Table 7.1: Probing results (in F_1) on the test set of BioRED and Re-DocRED.

straightforward finetuning method is to add a softmax classifier upon the relation embedding, for which a cross-entropy loss for a batch of relation mentions $\mathscr{T}$ is formulated as:

$$P_r^{t_i} = \text{softmax}(\boldsymbol{W}_r \boldsymbol{z}^{t_i} + b_r),$$
$$\mathscr{L}_{\text{ce}} = -\frac{1}{|\mathscr{T}|} \sum_{t_i \in \mathscr{T}} \log(P_{y_{t_i}}^{t_i}),$$

where y_t is the ground-truth label for relation mention t, $\boldsymbol{W}_r, b_r$ are the weight and bias of the classifier. Though this approach has shown improvements, it may produce sub-optimal outcomes from MTB pretraining since it implicitly assumes that relation mentions corresponding to the same relation are in the same cluster, while MTB pretraining may learn multiple clusters for a relation. For example, the entity pairs *(Honda Corp., Japan)* and *(Mount Fuji, Japan)*, although likely to be expressed with the same relation *country* in documents, are likely to be in different clusters since MTB views them as negative pairs due to different entity names. Therefore, we propose an MCCL objective that can bridge these gaps. Next, we will discuss the distributional assumption of the softmax classifier as well as supervised contrastive loss, then present our MCCL objective.

Distributional assumption. We conduct a probing analysis on the distribution of pretrained representations to further justify the multi-cluster assumption. Specifically, we fix the weights of the pretrained MTB model and fit different classifiers on top of it, including a softmax classifier, nearest centroid classifier (both assuming one cluster for a relation), and kNN classifier (assuming multiple clusters for a relation). We evaluate these classifiers on the test set. Results are shown in Table 7.1. We find that kNN greatly outperforms other classifiers, showing that MTB pretraining learns multiple clusters for a relation.

Therefore, to accommodate this multi-cluster assumption, we need to finetune the representations with a training objective that suits multiple clusters for each relation. Besides using the softmax classifier with cross-entropy loss, we also consider supervised contrastive loss (SupCon; [154, 155]), which is shown to be more effective than softmax on NLP tasks. SupCon has a similar loss form to InfoNCE in Eq. (7.1), except that it uses instances of the same/different relations as positive/negative pairs. However, previous work [261] has shown that both softmax and SupCon are minimized when the representations of each class collapse to the vertex of a regular simplex. In our case, this means the relation embedding corresponding to the same relation in pretraining collapses to a single point, which creates a distributional gap between pretraining and finetuning.

Training objective. We thereby propose the MCCL objective. Given relation mentions $\mathscr{T}$ and sets of relation mentions grouped by their relations $\{\mathscr{T}_r\}_{r\in\mathscr{R}}$, our loss is formulated as:

$$
\begin{aligned}
w_r^{(t_i,t_j)} &= \frac{e^{\mathrm{sim}(\boldsymbol{z}^{t_i},\boldsymbol{z}^{t_j})/\tau_1}}{\sum_{t_k\in\mathscr{T}_r\setminus\{t_i\}} e^{\mathrm{sim}(\boldsymbol{z}^{t_i},\boldsymbol{z}^{t_k})/\tau_1}}, \\
s_r^{t_i} &= \sum_{t_j\in\mathscr{T}_r\setminus\{t_i\}} w_r^{(t_i,t_j)}\,\mathrm{sim}(\boldsymbol{z}^{t_i},\boldsymbol{z}^{t_j}), \\
P_r^{t_i} &= \mathrm{softmax}((s_r^{t_i}+b_r)/\tau_2), \\
\mathscr{L}_{\mathrm{mccl}} &= -\frac{1}{|\mathscr{T}|}\sum_{t_i\in\mathscr{T}} \log(P_{y_{t_i}}^{t_i}),
\end{aligned}
$$

where τ_1 and τ_2 are temperature hyperparameters, $b_r\in\mathbb{R}$ is the classwise bias. The loss calculation can be split into two parts. First, we calculate the similarity between t_i and relation r, which is a weighted average of the similarity between t_i and $t_j\in\mathscr{T}_r$ such that a more similar t_j has a larger weight. Next, we use the cross-entropy loss to make the similarity of ground-truth relation larger than others. In this way, MCCL only optimizes t_i to be similar to a few closest relation mentions of the ground-truth relation, and thus encourages multiple clusters in relation embedding.

Proxies. We use batched training for finetuning, where relation mentions in the current batch are used to calculate MCCL. However, it is possible that a subset of relations in $\mathscr{R}$, especially the long-tail relations, are rare or missing in the current batch. When $\mathscr{T}_r\setminus\{t_i\}$ is empty, $s_r^{t_i}$ and the MCCL loss become undefined. To tackle this problem, we propose the use of proxies [279, 280]. We add one proxy vector p_r for each relation r, which is a trainable parameter and associated with an embedding $\boldsymbol{z}_r^p$. We incorporate the proxies into MCCL by changing $\mathscr{T}_r$ to $\mathscr{T}_r'=\mathscr{T}_r\cup\{p_r\}$, ensuring that $\mathscr{T}_r'\setminus\{t_i\}$ is never empty in training and preventing MCCL from becoming undefined. The proxies are randomly initialized and updated during training by backward propagation.

7.3.4 Inference

We use the classwise kNN [281] for inference, which predicts relations based on similarly represented instances and thus aligns with our contrastive finetuning objective. Given a new relation mention to predict, we first find k most similar instances[3] in the training data of each relation (including NA), then calculate the average cosine similarity of each relation s_r^{avg}. Finally, the model returns the relation with the maximum $s_r^{\mathrm{avg}}+b_r$ for single-label prediction, and all relations with higher $s_r^{\mathrm{avg}}+b_r$ than NA for multi-label prediction. We use classwise kNN because it is more suitable for RE datasets, where the label distribution is usually long-tailed [282].

7.4 Experiments

We evaluate our proposed method with a focus on low-resource RE (Sections 7.4.1-7.4.3), and present detailed analyses (Section 7.4.4) and visualization (Section 7.4.5) to justify method design choices.

[3]Measured by cosine similarity. If a relation has fewer than k mentions in training data, we use all of its mentions.

7.4.1 Datasets

We conduct experiments with two document-level RE datasets. The **BioRED** dataset [11] is a manually labeled single-label RE dataset in the biomedical domain. The relation mentions are classified into 9 types (including an NA type indicating no relation). It has a training set consisting of 400 documents, which we use in finetuning. For pretraining, we use the PubTator Central corpus [283], which annotates the PubMed corpus with entity mentions and their named entity types. The **Re-DocRED** dataset [265] is a multi-label large-scale dataset of the general domain. It is a relabeled version of the DocRED dataset [21]. Re-DocRED addresses the incomplete annotation issue of DocRED, where a large percentage of relation mentions are mislabeled as NA. The relation mentions in Re-DocRED are classified into 97 types (incl. NA). It has a training set consisting of 3,053 documents, which we use in finetuning. For pretraining, we use the distantly labeled training set provided by DocRED, which consists of 101,873 documents. We remove the relation labels and use our improved MTB to pretrain the model.

7.4.2 Experimental Setup

Model configurations. We implement our models using Hugging Face Transformers [167]. We use AdamW [284] in optimization with a weight decay of 0.01. During pretraining, we use a batch size of 16, a learning rate of 5e-6, a temperature of 0.05, and epochs of 3 and 10 for BioRED and DocRED, respectively. During finetuning, we use a batch size of 32, a learning rate of 5e-5, and epochs of 100 and 30 for BioRED and DocRED, respectively. The temperatures in MCCL are set to $\tau_1 = \tau_2 = 0.2$ for BioRED and $\tau_1 = 0.01, \tau_2 = 0.03$ for DocRED. We search k from $\{1,3,5,10,20\}$ for kNN using the development set[4]. We run experiments with Nvidia V100 GPUs.

Evaluation settings. In this work, in addition to the standard full-shot training, we consider low-resource settings. To create each of the settings, we randomly sample a fixed proportion $p\%$ of the relation mentions from the training set as our training data, and use the original test set for evaluation. We use the same evaluation metrics as the original papers. We use micro-F_1 for BioRED, and micro-F_1 and micro-F_1-Ign for Re-DocRED. The micro-F_1-Ign removes the relational facts in the test set that have appeared in training.

Compared methods. We experiment with the following finetuning objectives: (1) **Lazy learning**, which directly uses the pretrained embedding and training data to perform kNN without finetuning; (2) **Cross-entropy loss** (CE), which adds a softmax classifier on top of PLM and uses cross-entropy loss to finetune the model; (3) **Supervised contrastive loss** (SupCon); and (4) **Multi-center contrastive loss** (MCCL). In inference, classwise kNN is used for all methods except for CE. Note that as SupCon does not apply to multi-label scenarios, we only evaluate it on BioRED. For each objective, we also evaluate the PLM before and after MTB pretraining. We use different PLMs as the backbone of the model, namely PubmedBERT$_{\text{BASE}}$ for BioRED and BERT$_{\text{BASE}}$ for Re-DocRED, which are pretrained on the biomedical and general domains, respectively.

[4]For low-resource setting with $p\%$ of training data, we sample $p\%$ of development data as the development set.

Encoder	Objective	1%		5%		10%		100%	
		F_1	F_1-Ign	F_1	F_1-Ign	F_1	F_1-Ign	F_1	F_1-Ign
PLM	Lazy	15.6	14.9	20.1	19.4	21.6	19.2	28.7	28.0
	CE	40.3	38.9	54.1	52.6	61.3	60.3	70.9	69.4
	MCCL	44.7	43.1	59.1	57.5	63.2	61.8	68.2	66.7
MTB	Lazy	35.2	34.4	44.7	43.4	47.3	46.2	54.1	52.9
	CE	42.3	40.7	57.9	56.4	62.9	61.4	**71.2**	**69.9**
	MCCL	**46.4**	**44.5**	**59.7**	**58.2**	**63.8**	**62.1**	69.3	67.9

Table 7.2: Results on the test set of Re-DocRED.

Encoder	Objective	1%	5%	10%	100%
PLM	Lazy	14.5	17.6	18.8	28.3
	CE	24.1	35.4	42.5	57.7
	SupCon	20.0	30.9	38.0	52.2
	MCCL	20.8	41.3	45.5	55.1
MTB	Lazy	24.3	28.4	34.4	36.7
	CE	28.6	41.2	49.8	**61.5**
	SupCon	24.4	29.1	31.4	43.1
	MCCL	**34.6**	**48.5**	**54.2**	60.8

Table 7.3: F_1 on the test set of BioRED.

7.4.3 Main Results

The results on the test sets of Re-DocRED and BioRED are shown in Table 7.2 and Table 7.3, respectively. All results are averaged for five runs of training using different random seeds. Overall, the combination of MTB and MCCL achieves the best performance in low-resource settings where 1%, 5%, and 10% of relation-labeled data are used. Further, when using the same MTB-based representations, MCCL shows better results than CE in low-resource settings. It shows that in low-resource settings, MCCL can better leverage the pretraining knowledge with a well-aligned finetuning objective. However, this improvement diminishes when abundant labeled data are available, as MCCL underperforms CE on both datasets with full training data on both datasets. In addition, we observe that MTB pretraining consistently improves MCCL and CE on both datasets. These results demonstrate the effectiveness of MTB pretraining for more precise document-level RE with less needed end-task supervision.

Considering other training objectives, we observe that lazy learning produces meaningful results. On both datasets, the results of lazy learning based on MTB with 10% of data are comparable to finetuning with 1% of data. This shows that the relation embedding pretrained on unlabeled corpora contains knowledge that can be transferred to unseen relations. We also observe that SupCon using kNN-based inference underperforms both CE and MCCL on BioRED, showing that its one-cluster assumption hurts the knowledge transfer.

Pretraining Objective	1%	10%	100%
PLM	20.8	45.5	55.1
vanilla MTB	22.9	45.0	56.0
our MTB	34.6	54.2	60.8
w/o $\mathscr{L}_{\text{rel}}$	21.0	47.1	56.7
w/o $\mathscr{L}_{\text{self}}$	24.1	49.3	58.6
w/o $\mathscr{L}_{\text{mlm}}$	32.9	50.2	58.2

Table 7.4: F_1 on the test set of BioRED with different pretraining objectives. We use MCCL in finetuning.

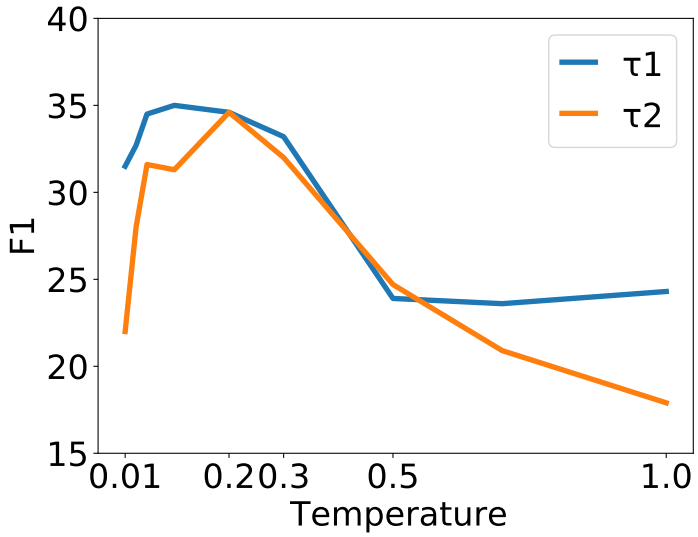

Figure 7.1: F_1 using different temperatures on 1% of BioRED.

7.4.4 Ablation Study

Pretraining objectives. We analyze the effectiveness of our proposed pretraining losses in Section 7.3.2. To do so, we pretrain the model with one loss removed at a time while keeping the finetuning setup on BioRED fixed with the MCCL loss. The results are shown in Table 7.4. Overall, we observe that all losses are effective. If we remove all proposed techniques and use the vanilla MTB pretraining objective of binary pairwise classification, the results are only slightly better or even worse. Among the techniques, removing $\mathscr{L}_{\text{rel}}$ leads to the largest performance drop, showing that MTB-based pretraining is critical to improve low-resource RE. Removing $\mathscr{L}_{\text{self}}$ also leads to a large performance drop. It is because $\mathscr{L}_{\text{self}}$ encourages the model to learn more discriminative features that lead to less collapsed representations. Our finding aligns with recent studies in computer vision [285, 286], showing that reducing collapsed representations with self-supervised contrastive learning improves the transferability to downstream tasks.

Performance w.r.t. different temperatures. We discuss the impact of two temperatures in MCCL. In MCCL, τ_1 controls the weighting of instances. With a very small τ_1, each instance will only form a cluster with its nearest neighbor in the batch, while with very large τ_1, instances of the same

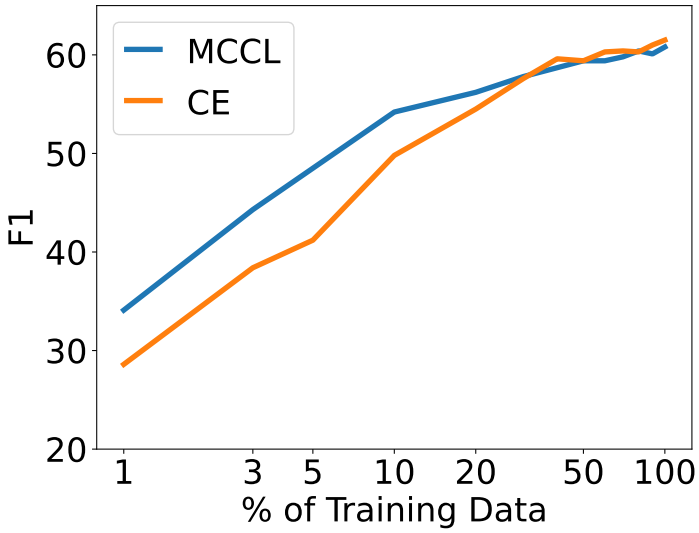

Figure 7.2: F_1 under different percentages of BioRED training data.

relation will collapse to the same cluster. τ_2 controls the importance of hard instances, which is also used in other contrastive losses (e.g., τ in Eq. (7.1)). Wang & Liu [287] observe that small τ_2 makes the model focus more on hard instances, while Khosla, Teterwak, Wang, Sarna, Tian, Isola, *et al.* [154] observe that too small τ_2 leads to numerical instability. We show the results of using different temperatures in Figure 7.1, where we keep one temperature fixed and change the other. For τ_1, we find that using large temperature harms the performance significantly, showing that our multi-cluster assumption improves low-resource RE. For τ_2, we observe that both small and large values impair the performance, which is aligned with prior observations.

Performance w.r.t. different amount of data. The main results show that MCCL outperforms CE in the low-resource setting, while slightly underperforming CE when full training data is used. We further evaluate MCCL and CE using different amounts of end-task data. We experiment on BioRED and use the relation embedding pretrained with MTB. Results are shown in Figure 7.2. We observe that MCCL consistently outperforms CE by a large marge when less than 20% of training data is used, while it performs similarly or worse than CE after that. It again demonstrates the effectiveness of MCCL in low-resource RE. However, as the pretraining and finetuning are based on different tasks, fully adapting the model to downstream data by CE results in similar or better performance in data-sufficient scenarios.

7.4.5 Visualization

Figure 7.3 shows the t-SNE [168] projection of relation embedding finetuned with different objectives on BioRED. For clarity, we visualize the embedding of the four most frequent relations in BioRED with different colors, including the NA class shown in grey. The visualization shows that both CE and SupCon learn one cluster for each relation, while lazy learning and MCCL, as expected, generate multiple small clusters for a relation. This observation indicates that MCCL can better align with the pretraining objective, further explaining its better performance in low-resource settings.

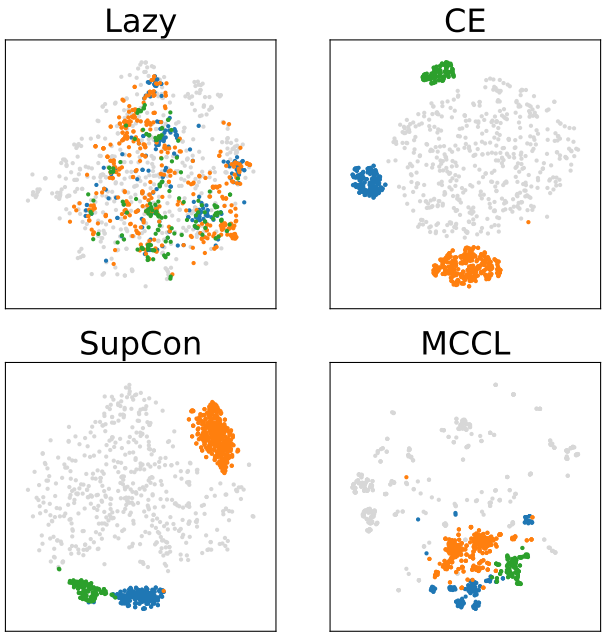

Figure 7.3: Visualization of relation embedding finetuned with different objectives on BioRED. NA instances are shown in grey.

7.5 Conclusion

We study self-supervised learning for document-level RE. Our method conducts an improved MTB pretraining objective that acquires cheap supervision signals from large corpora without relation labels. To bridge the gap between pretraining and end-task finetuning, we propose a continual contrastive finetuning objective, in contrast to prior studies that typically use classification-based finetuning, and use kNN-based inference. As pretrained representation may form multi-cluster representation, we further propose a multi-center contrastive loss that aligns well with the nature of the pretrained representation. Extensive experiments on two document-level RE datasets demonstrate the effectiveness of these key techniques in our method. Future work is adapting our method to other tasks in information extraction, such as n-ary relation extraction, named entity recognition, typing, and linking.

Chapter 8

Conclusion and Future Work

8.1 Summary

In this dissertation, we aim to build robust and generalizable knowledge acquisition systems. Our contributions can be summarized into three aspects:

1. We propose a framework for extracting knowledge from long contexts such as documents.

2. We study the robustness of knowledge acquisition systems from multiple aspects, including robustness to noisy training labels, out-of-distribution instances, and knowledge conflicts.

3. We investigate methods for building data-efficient knowledge acquisition systems. This includes data annotation by neural rule grounding, as well as improving data efficiency through continual contrastive finetuning.

More specifically, in Chapter 2, we proposed the ATLOP model for document-level relation extraction, featuring two techniques, namely adaptive thresholding and localized context pooling. Adaptive thresholding learned a threshold class that decides the best threshold for each entity pair. Localized context pooling utilized attention in PLMs to locate salient contexts for entity pairs.

In Chapter 3, we proposed a co-regularization framework for learning from noisy training labels. This framework involved multiple models with identical structures but different initializations, optimizing an agreement loss to encourage them to give similar predictions on the same inputs. On noisy examples, where model predictions often deviate from the labels, the agreement loss prevented overfitting on noisy labels.

In Chapter 4, we presented an unsupervised OOD detection framework for pretrained Transformers. We systematically investigated the combination of contrastive losses and scoring functions. Specifically, we proposed a margin-based contrastive objective for learning compact representations that, when combined with the Mahalanobis distance, achieved near-perfect OOD detection performance on various tasks and datasets.

In Chapter 5, we addressed the faithfulness issue of LLMs, particularly in scenarios with knowledge conflict and prediction with abstention. We proposed two effective methods to enhance LLMs' faithfulness to contexts: opinion-based prompts and counterfactual demonstrations, which significantly improve the faithfulness of LLMs to contexts.

In Chapter 6, we proposed the NERO framework for label-efficient relation extraction. The framework involved automatically extracting frequent patterns from large raw corpora and requesting

users to create labeling rules. To improve the coverage of these rules, we proposed a soft-matching mechanism, where unlabeled sentences are annotated using their most semantically similar labeling rules and weighted when training the RE model.

Finally, in Chapter 7, we proposed a self-supervised framework to utilize unlabeled data in knowledge acquisition. We pretrained the model using an improved MTB pretraining objective. To address the gap between pretraining and end-task finetuning, we proposed a continual contrastive finetuning objective and kNN-based inference. We further proposed a multi-center contrastive loss that aligns well with the nature of the pretrained representation.

This dissertation thoroughly addresses several challenges of knowledge acquisition in real-world applications. The methods proposed in this research support a wide range of applications, including but not limited to constructing knowledge bases, handling knowledge-intensive NLP tasks such as question answering, fact verification, and dialogue, as well as biomedical and clinical applications such as in-silico biomedical research and precision medicine.

8.2 Future Directions

We discuss several promising future directions.

- **Universal knowledge acquisition.** Most current knowledge acquisition systems are designed for specific tasks (such as entities, events, etc.) or domains (such as general, biomedical, finance, etc.). Unifying these domains and tasks offers two significant advantages. First, many knowledge acquisition tasks require common information, such as syntactic features like dependencies, which are essential for both entity [30] and event-centric relation extraction [288]. In addition, the outputs of entity or event detection tasks are typically used as inputs for relation extraction tasks. As such, unifying different knowledge acquisition tasks can improve the ability to learn and utilize shared cross-task information. Second, by jointly learning across multiple domains, a knowledge acquisition system can learn transferable features that generalize to different domains, resulting in improved domain generalization ability. As a result, universal knowledge acquisition has the potential to perform any task in any distribution in a zero-shot manner, which can significantly benefit real-world applications, particularly in high-stakes domains.

- **Cross-document knowledge acquisition.** In addition to relations within a single document, some relations may require complex reasoning over multiple documents. As highlighted in a recent quantitative study [289], over 57.6% of relational facts in Wikidata involve subject and object entities that do not appear in the same Wikipedia document. Therefore, cross-document knowledge acquisition represents a more practical direction for developing practical knowledge acquisition systems. This problem also poses new research challenges, such as retrieving salient contexts from multiple documents and performing multi-step cross-document relation reasoning.

- **Fact-checking for large language models.** Large language models such as ChatGPT are increasingly popular in natural language processing. However, their responses may contain factual errors despite being seemingly plausible [290]. Therefore, fact-checking the outputs of these models is critical for ensuring AI safety. To achieve this, knowledge acquisition

systems can be used to convert the models' responses into structured formats, making them easier to align and verify against existing knowledge bases. This approach can enhance the accuracy of language models and reduce the potential harm caused by misinformation, ultimately advancing the development of responsible and reliable AI systems.

Bibliography

1. Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R. & Song, D. *Pretrained Transformers Improve Out-of-Distribution Robustness* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, 2020), 2744–2751.

2. Zhang, Y., Dai, H., Kozareva, Z., Smola, A. & Song, L. *Variational reasoning for question answering with knowledge graph* in *Proceedings of the AAAI conference on artificial intelligence* **32** (2018).

3. Huang, X., Zhang, J., Li, D. & Li, P. *Knowledge graph embedding based question answering* in *Proceedings of the twelfth ACM international conference on web search and data mining* (2019), 105–113.

4. Zhou, W., Ning, Q., Elfardy, H., Small, K. & Chen, M. *Answer Consolidation: Formulation and Benchmarking* in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (eds Carpuat, M., de Marneffe, M.-C. & Meza Ruiz, I. V.) (Association for Computational Linguistics, Seattle, United States, 2022), 4314–4325. doi:10.18653/v1/2022.naacl-main.320.

5. Shiralkar, P., Flammini, A., Menczer, F. & Ciampaglia, G. L. *Finding streams in knowledge graphs to support fact checking* in *2017 IEEE International Conference on Data Mining (ICDM)* (2017), 859–864.

6. Zhou, J., Han, X., Yang, C., Liu, Z., Wang, L., Li, C., *et al. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, 2019), 892–901. doi:10.18653/v1/P19-1085.

7. Eric, M., Krishnan, L., Charette, F. & Manning, C. D. *Key-Value Retrieval Networks for Task-Oriented Dialogue* in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue* (Association for Computational Linguistics, Saarbrücken, Germany, 2017), 37–49. doi:10.18653/v1/W17-5506.

8. Yang, S., Zhang, R. & Erfani, S. *GraphDialog: Integrating Graph Knowledge into End-to-End Task-Oriented Dialogue Systems* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 1878–1888. doi:10.18653/v1/2020.emnlp-main.147.

9. Herrero-Zazo, M., Segura-Bedmar, I., Martınez, P. & Declerck, T. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics* **46,** 914–920 (2013).

10. Singhal, A., Simmons, M. & Lu, Z. Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. *PLoS computational biology* **12,** e1005017 (2016).

11. Luo, L., Lai, P.-T., Wei, C.-H., Arighi, C. N. & Lu, Z. BioRED: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics* **23,** bbac282 (2022).

12. Lv, X., Guan, Y., Yang, J. & Wu, J. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology* **9,** 237–248 (2016).

13. Wei, Q., Ji, Z., Si, Y., Du, J., Wang, J., Tiryaki, F., *et al. Relation extraction from clinical narratives using pre-trained language models* in *AMIA annual symposium proceedings* **2019** (2019), 1236.

14. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. *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)* (Minneapolis, Minnesota, 2019), 4171–4186.

15. Alt, C., Hübner, M. & Hennig, L. *Improving Relation Extraction by Pre-trained Language Representations* in *AKBC* (2019).

16. Shi, P. & Lin, J. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *ArXiv* **abs/1904.05255** (2019).

17. Zhou, W. & Chen, M. *An Improved Baseline for Sentence-level Relation Extraction* in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Association for Computational Linguistics, Online only, 2022), 161–168.

18. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., *et al. DocRED: A Large-Scale Document-Level Relation Extraction Dataset* in *ACL* (2019).

19. Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., *et al. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals* in *Proceedings of the 5th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Uppsala, Sweden, 2010), 33–38.

20. Zhang, Y., Zhong, V., Chen, D., Angeli, G. & Manning, C. D. *Position-aware Attention and Supervised Data Improve Slot Filling* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), 35–45. doi:10.18653/v1/D17-1004.

21. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., *et al. DocRED: A Large-Scale Document-Level Relation Extraction Dataset* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, 2019), 764–777. doi:10.18653/v1/P19-1074.

22. Mintz, M., Bills, S., Snow, R. & Jurafsky, D. *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* (Association for Computational Linguistics, Suntec, Singapore, 2009), 1003–1011.

23. Zeng, D., Liu, K., Chen, Y. & Zhao, J. *Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks* in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Lisbon, Portugal, 2015), 1753–1762.

24. Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., *et al. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Brussels, Belgium, 2018), 4803–4809. doi:10.18653/v1/D18-1514.

25. Sainz, O., Lopez de Lacalle, O., Labaka, G., Barrena, A. & Agirre, E. *Label Verbalization and Entailment for Effective Zero and Few-Shot Relation Extraction* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 1199–1212. doi:10.18653/v1/2021.emnlp-main.92.

26. Patterson, S. E., Liu, R., Statz, C. M., Durkin, D., Lakshminarayana, A. & Mockus, S. M. The clinical trial landscape in oncology and connectivity of somatic mutational profiles to targeted therapies. *Human genomics* **10,** 1–13 (2016).

27. Zhou, W., Huang, K., Ma, T. & Huang, J. *Document-level relation extraction with adaptive thresholding and localized context pooling* in *Proceedings of the AAAI conference on artificial intelligence* **35** (2021), 14612–14620.

28. Zeng, D., Liu, K., Lai, S., Zhou, G. & Zhao, J. *Relation Classification via Convolutional Deep Neural Network* in *COLING* (2014).

29. Miwa, M. & Bansal, M. *End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures* in *ACL* (2016).

30. Zhang, Y., Qi, P. & Manning, C. D. *Graph Convolution over Pruned Dependency Trees Improves Relation Extraction* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Brussels, Belgium, 2018), 2205–2215. doi:10.18653/v1/D18-1244.

31. Verga, P., Strubell, E. & McCallum, A. *Simultaneously Self-Attending to All Mentions for Full-Abstract Biological Relation Extraction* in *NAACL-HLT* (2018).

32. Peng, N., Poon, H., Quirk, C., Toutanova, K. & Yih, W.-t. Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics* **5,** 101–115 (2017).

33. Liu, Y. & Lapata, M. Learning Structured Text Representations. *Transactions of the Association for Computational Linguistics* **6,** 63–75 (2018).

34. Christopoulou, F., Miwa, M. & Ananiadou, S. *Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs* in *EMNLP-IJCNLP* (2019).

35. Nan, G., Guo, Z., Sekulic, I. & Lu, W. *Reasoning with Latent Structure Refinement for Document-Level Relation Extraction* in *ACL* (2020).

36. Liang, X., Shen, X., Feng, J., Lin, L. & Yan, S. *Semantic Object Parsing with Graph LSTM* in *ECCV* (2016).

37. Guo, Z., Zhang, Y. & Lu, W. *Attention Guided Graph Convolutional Networks for Relation Extraction* in *ACL* (2019).

38. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9,** 1735–1780 (1997).

39. Chung, J., Gülçehre, Ç., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv* **abs/1412.3555** (2014).

40. Khandelwal, U., He, H., Qi, P. & Jurafsky, D. *Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context* in *ACL* (2018).

41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., *et al. Attention is All you Need* in *NeurIPS* (2017).

42. Clark, K., Khandelwal, U., Levy, O. & Manning, C. D. *What Does BERT Look at? An Analysis of BERT's Attention* in *BlackboxNLP workshop* (2019).

43. Tenney, I., Das, D. & Pavlick, E. *BERT Rediscovers the Classical NLP Pipeline* in *ACL* (2019).

44. Wang, H., Focke, C., Sylvester, R., Mishra, N. & Wang, W. W. J. Fine-tune Bert for DocRED with Two-step Process. *ArXiv* **abs/1909.11898** (2019).

45. Tang, H., Cao, Y., Zhang, Z., Cao, J., Fang, F., Wang, S., *et al. HIN: Hierarchical Inference Network for Document-Level Relation Extraction* in *PAKDD* (2020).

46. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., *et al.* Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

47. Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., *et al. BioCreative V CDR task corpus: a resource for chemical disease relation extraction* in *Database* (2016).

48. Wu, Y., Luo, R., Leung, H. C., Ting, H.-F. & Lam, T.-W. *RENET: A Deep Learning Approach for Extracting Gene-Disease Associations from Literature* in *RECOMB* (2019).

49. Soares, L. B., FitzGerald, N., Ling, J. & Kwiatkowski, T. *Matching the Blanks: Distributional Similarity for Relation Learning* in *ACL* (2019).

50. Wang, H., Tan, M., Yu, M., Chang, S., Wang, D., Xu, K., *et al. Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers* in *ACL* (2019).

51. Jia, R., Wong, C. & Poon, H. *Document-Level N-ary Relation Extraction with Multiscale Representation Learning* in *NAACL-HLT* (2019).

52. Zheng, H., Fu, J., Zha, Z.-J. & Luo, J. *Learning Deep Bilinear Transformation for Fine-grained Image Representation* in *NeurIPS* (2019).

53. Tang, Y., Huang, J., Wang, G., He, X. & Zhou, B. *Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding* in *ACL* (2020).

54. Menon, A., Rawat, A., Reddi, S. & Kumar, S. *Multilabel reductions: what is my loss optimising?* in *NeurIPS* (2019).

55. Reddi, S. J., Kale, S., Yu, F., Holtmann-Rice, D., Chen, J. & Kumar, S. *Stochastic Negative Mining for Learning with Large Output Spaces* in *AISTATS* (2019).

56. Ye, D., Lin, Y., Du, J., Liu, Z., Sun, M. & Liu, Z. *Coreferential Reasoning Learning for Language Representation* in *EMNLP* (2020).

57. Nguyen, D. Q. & Verspoor, K. *Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings* in *BioNLP workshop* (2018).

58. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., *et al.* HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv,* arXiv–1910 (2019).

59. Beltagy, I., Lo, K. & Cohan, A. *SciBERT: A Pretrained Language Model for Scientific Text* in *EMNLP-IJCNLP* (2019).

60. Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., García, D., *et al. Mixed Precision Training* in *ICLR* (2018).

61. Loshchilov, I. & Hutter, F. *Decoupled Weight Decay Regularization* in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (OpenReview.net, 2019).

62. Goyal, P., Dollár, P., Girshick, R. B., Noordhuis, P., Wesolowski, L., Kyrola, A., *et al.* Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *ArXiv* **abs/1706.02677** (2017).

63. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15,** 1929–1958 (2014).

64. Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. *Deep learning* (MIT press Cambridge, 2016).

65. Schuster, M. & Paliwal, K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45,** 2673–2681 (1997).

66. Kipf, T. N. & Welling, M. *Semi-Supervised Classification with Graph Convolutional Networks* in *International Conference on Learning Representations (ICLR)* (2017).

67. Fan, R.-E. & Lin, C.-J. A study on threshold selection for multi-label classification. *Department of Computer Science, National Taiwan University,* 1–23 (2007).

68. Wang, Z., Qin, Y., Zhou, W., Yan, J., Ye, Q., Neves, L., *et al.* Learning from explanations with neural execution tree. *arXiv preprint arXiv:1911.01352* (2019).

69. Zhou, W., Lin, H., Lin, B. Y., Wang, Z., Du, J., Neves, L., *et al. Nero: A neural rule grounding framework for label-efficient relation extraction* in *Proceedings of The Web Conference 2020* (2020), 2166–2176.

70. Wang, L., Cao, Z., De Melo, G. & Liu, Z. *Relation Classification via Multi-Level Attention CNNs* in *ACL* (2016).

71. Wu, F., Zhang, T., Souza, A., Fifty, C., Yu, T. & Weinberger, K. Q. *Simplifying Graph Convolutional Networks* in *ICML* (2019).

72. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M. & Liu, Q. *ERNIE: Enhanced Language Representation with Informative Entities* in *ACL* (2019).

73. Quirk, C. & Poon, H. *Distant Supervision for Relation Extraction beyond the Sentence Boundary* in *EACL* (2017).

74. Song, L., Zhang, Y., Wang, Z. & Gildea, D. *N-ary Relation Extraction using Graph-State LSTM* in *EMNLP* (2018).

75. Gupta, P., Rajaram, S., Schütze, H. & Runkler, T. *Neural Relation Extraction Within and Across Sentence Boundaries* in *AAAI* (2019).

76. Wang, Y., Chen, M., Zhou, W., Cai, Y., Liang, Y. & Hooi, B. *GraphCache: Message Passing as Caching for Sentence-Level Relation Extraction* in *Findings of the Association for Computational Linguistics: NAACL 2022* (eds Carpuat, M., de Marneffe, M.-C. & Meza Ruiz, I. V.) (Association for Computational Linguistics, Seattle, United States, 2022), 1698–1708. doi:10.18653/v1/2022.findings-naacl.128.

77. Christopoulou, F., Miwa, M. & Ananiadou, S. *A Walk-based Model on Entity Graphs for Relation Extraction* in *ACL* (2018).

78. Vig, J. & Belinkov, Y. *Analyzing the Structure of Attention in a Transformer Language Model* in *BlackboxNLP workshop* (2019).

79. Hewitt, J. & Manning, C. D. *A Structural Probe for Finding Syntax in Word Representations* in *NAACL-HLT* (2019).

80. Zhou, W. & Chen, M. *Learning from Noisy Labels for Entity-Centric Information Extraction* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (eds Moens, M.-F., Huang, X., Specia, L. & Yih, S. W.-t.) (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 5381–5392. doi:10.18653/v1/2021.emnlp-main.437.

81. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., *et al. A Closer Look at Memorization in Deep Networks* in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017* (eds Precup, D. & Teh, Y. W.) **70** (PMLR, 2017), 233–242.

82. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. *Understanding deep learning requires rethinking generalization* in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net, 2017).

83. Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., *et al.* Learning from crowds. *Journal of Machine Learning Research* **11** (2010).

84. Song, Y., Wang, C., Zhang, M., Sun, H. & Yang, Q. *Spectral Label Refinement for Noisy and Missing Text Labels* in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA* (eds Bonet, B. & Koenig, S.) (AAAI Press, 2015), 2972–2978.

85. Sang, E. T. K. & De Meulder, F. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition* in *Proceedings of CoNLL-2003, Edmonton, Canada* (2003), 142–145.

86. Reiss, F., Xu, H., Cutler, B., Muthuraman, K. & Eichenberger, Z. *Identifying Incorrect Labels in the CoNLL-2003 Corpus* in *Proceedings of the 24th Conference on Computational Natural Language Learning* (Association for Computational Linguistics, Online, 2020), 215–226.

87. Alt, C., Gabryszak, A. & Hennig, L. *TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Online, 2020), 1558–1569.

88. Surdeanu, M., Tibshirani, J., Nallapati, R. & Manning, C. D. *Multi-instance Multi-label Learning for Relation Extraction* in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Association for Computational Linguistics, Jeju Island, Korea, 2012), 455–465.

89. Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D. & Ré, C. *Data programming: Creating large training sets, quickly* in *Advances in neural information processing systems* (2016), 3567–3575.

90. Huang, Y. & Du, J. *Self-Attention Enhanced CNNs and Collaborative Curriculum Learning for Distantly Supervised Relation Extraction* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 389–398.

91. Mayhew, S., Chaturvedi, S., Tsai, C.-T. & Roth, D. *Named Entity Recognition with Partially Annotated Training Data* in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (Association for Computational Linguistics, Hong Kong, China, 2019), 645–655.

92. Qin, P., Xu, W. & Wang, W. Y. *Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Melbourne, Australia, 2018), 2137–2147.

93. Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J. & Han, J. *CrossWeigh: Training Named Entity Tagger from Imperfect Annotations* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 5154–5163.

94. Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y. & Gordon, G. J. *An Empirical Study of Example Forgetting during Deep Neural Network Learning* in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019* (OpenReview.net, 2019).

95. Jiang, L., Zhou, Z., Leung, T., Li, L. & Fei-Fei, L. *MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels* in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (eds Dy, J. G. & Krause, A.) **80** (PMLR, 2018), 2309–2318.

96. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., *et al. Co-teaching: Robust training of deep neural networks with extremely noisy labels* in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* (eds Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N. & Garnett, R.) (2018), 8536–8546.

97. Ramshaw, L. & Marcus, M. *Text Chunking using Transformation-Based Learning* in *Third Workshop on Very Large Corpora* (1995).

98. Yamada, I., Asai, A., Shindo, H., Takeda, H. & Matsumoto, Y. *LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 6442–6454.

99. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., *et al. Transformers: State-of-the-Art Natural Language Processing* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Association for Computational Linguistics, Online, 2020), 38–45.

100. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* in *International Conference for Learning Representations* (2015).

101. Lee, J. & Chung, S.-Y. *Robust training with ensemble consensus* in *International Conference on Learning Representations* (2020).

102. Zhou, W., Zhang, S., Gu, Y., Chen, M. & Poon, H. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279* (2023).

103. Lin, Y., Shen, S., Liu, Z., Luan, H. & Sun, M. *Neural Relation Extraction with Selective Attention over Instances* in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Berlin, Germany, 2016), 2124–2133.

104. Ji, G., Liu, K., He, S. & Zhao, J. *Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions* in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA* (eds Singh, S. P. & Markovitch, S.) (AAAI Press, 2017), 3060–3066.

105. Liu, T., Wang, K., Chang, B. & Sui, Z. *A Soft-label Method for Noise-tolerant Distantly Supervised Relation Extraction* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), 1790–1795.

106. Yang, K., He, L., Dai, X.-y., Huang, S. & Chen, J. *Exploiting Noisy Data in Distant Supervision Relation Classification* 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)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 3216–3225.

107. Wang, Z., Wen, R., Chen, X., Huang, S.-L., Zhang, N. & Zheng, Y. Finding Influential Instances for Distantly Supervised Relation Extraction. *ArXiv* **abs/2009.09841** (2020).

108. Zhang, Z. & Sabuncu, M. R. *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels* in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* (eds Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N. & Garnett, R.) (2018), 8792–8802.

109. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J. & Bailey, J. *Symmetric Cross Entropy for Robust Learning With Noisy Labels* in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019* (IEEE, 2019), 322–330.

110. Fang, T., Zhou, W., Liu, F., Zhang, H., Song, Y. & Chen, M. On-the-fly denoising for data augmentation in natural language understanding. *arXiv preprint arXiv:2212.10558* (2022).

111. Yang, X., Huang, J. Y., Zhou, W. & Chen, M. *Parameter-Efficient Tuning with Special Token Adaptation* in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (eds Vlachos, A. & Augenstein, I.) (Association for Computational Linguistics, Dubrovnik, Croatia, 2023), 865–872. doi:10.18653/v1/2023.eacl-main.60.

112. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L. D. & Fergus, R. *Training Convolutional Networks with Noisy Labels* in *ICLR* (2015).

113. Goldberger, J. & Ben-Reuven, E. *Training deep neural-networks using a noise adaptation layer* in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net, 2017).

114. Wang, R., Liu, T. & Tao, D. Multiclass learning with partially corrupted labels. *IEEE transactions on neural networks and learning systems* **29,** 2568–2580 (2017).

115. Chang, H.-S., Learned-Miller, E. & McCallum, A. *Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples* in *NeurIPS* (2017).

116. Krogh, A. & Hertz, J. A. *A simple weight decay can improve generalization* in *Advances in neural information processing systems* (1992), 950–957.

117. Müller, R., Kornblith, S. & Hinton, G. *When does label smoothing help?* in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (2019), 4694–4703.

118. Zhou, W., Lin, B. Y. & Ren, X. *Isobn: Fine-tuning bert with isotropic batch normalization* in *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (2021), 14621–14629.

119. Zhou, W., Liu, F., Zhang, H. & Chen, M. *Sharpness-Aware Minimization with Dynamic Reweighting* in *Findings of the Association for Computational Linguistics: EMNLP 2022* (eds Goldberg, Y., Kozareva, Z. & Zhang, Y.) (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), 5686–5699. doi:10.18653/v1/2022.findings-emnlp.417.

120. Malach, E. & Shalev-Shwartz, S. *Decoupling "when to update" from "how to update"* in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* (eds Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., *et al.*) (2017), 960–970.

121. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I. W. & Sugiyama, M. *How does Disagreement Help Generalization against Label Corruption?* in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (eds Chaudhuri, K. & Salakhutdinov, R.) **97** (PMLR, 2019), 7164–7173.

122. Wei, H., Feng, L., Chen, X. & An, B. *Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization* in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020* (IEEE, 2020), 13723–13732.

123. Wang, H., Liu, B., Li, C., Yang, Y. & Li, T. *Learning with Noisy Labels for Sentence-level Sentiment Classification* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 6286–6292.

124. Wang, R., Liu, T. & Tao, D. Multiclass Learning With Partially Corrupted Labels. *IEEE Transactions on Neural Networks and Learning Systems* **29,** 2568–2580 (2018).

125. Cheng, J., Liu, T., Ramamohanarao, K. & Tao, D. *Learning with Bounded Instance and Label-dependent Label Noise* in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event* **119** (PMLR, 2020), 1789–1799.

126. Chen, M., Zhang, H., Ning, Q., Li, M., Ji, H., McKeown, K., *et al. Event-Centric Natural Language Processing* in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics* (2021).

127. Peng, H., Khashabi, D. & Roth, D. *Solving Hard Coreference Problems* in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2015), 809–819.

128. Zhou, W., Liu, F. & Chen, M. *Contrastive Out-of-Distribution Detection for Pretrained Transformers* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 1100–1111. doi:10.18653/v1/2021.emnlp-main.84.

129. Hsu, Y.-C., Shen, Y., Jin, H. & Kira, Z. Generalized ODIN: Detecting Out-of-Distribution Image Without Learning From Out-of-Distribution Data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 10948–10957 (2020).

130. Hendrycks, D., Mazeika, M. & Dietterich, T. *Deep Anomaly Detection with Outlier Exposure* in *International Conference on Learning Representations* (2018).

131. Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., *et al. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019), 1311–1316.

132. Shu, L., Xu, H. & Liu, B. *DOC: Deep Open Classification of Text Documents* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), 2911–2916.

133. Dhamija, A., Günther, M. & Boult, T. *Reducing Network Agnostophobia* in *NeurIPS* (2018).

134. Liang, S., Li, Y. & Srikant, R. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks* in *International Conference on Learning Representations* (2018).

135. Sohn, K. *Improved deep metric learning with multi-class n-pair loss objective* in *Proceedings of the 30th International Conference on Neural Information Processing Systems* (2016), 1857–1865.

136. Oord, A. v. d., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

137. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. *A simple framework for contrastive learning of visual representations* in *International conference on machine learning* (2020), 1597–1607.

138. Lee, K., Lee, K., Lee, H. & Shin, J. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks* in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), 7167–7177.

139. Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J. & Mané, D. Concrete Problems in AI Safety. *ArXiv* **abs/1606.06565** (2016).

140. Lee, K., Lee, H., Lee, K. & Shin, J. *Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples* in *International Conference on Learning Representations* (2018).

141. Bergman, L. & Hoshen, Y. *Classification-Based Anomaly Detection for General Data* in *International Conference on Learning Representations* **abs/2005.02359** (2020).

142. Bendale, A. & Boult, T. Towards Open Set Deep Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1563–1572 (2016).

143. Hendrycks, D. & Gimpel, K. *A baseline for detecting misclassified and out-of-distribution examples in neural networks* in *International Conference on Learning Representations* (2017).

144. Liu, W., Wang, X., Owens, J. & Li, Y. *Energy-based Out-of-distribution Detection* in *Advances in Neural Information Processing Systems* **33** (2020).

145. Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B. & Willke, T. L. *Out-of-distribution detection using an ensemble of self supervised leave-out classifiers* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 550–564.

146. Kim, J.-K. & Kim, Y.-B. Joint Learning of Domain Classification and Out-of-Domain Detection with Dynamic Class Weighting for Satisficing False Acceptance Rates. *Proc. Interspeech 2018,* 556–560 (2018).

147. Tan, M., Yu, Y., Wang, H., Wang, D., Potdar, S., Chang, S., *et al. Out-of-Domain Detection for Low-Resource Text Classification Tasks* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, 2019), 3566–3572.

148. Wang, T. & Isola, P. *Understanding contrastive representation learning through alignment and uniformity on the hypersphere* in *International Conference on Machine Learning* (2020), 9929–9939.

149. Misra, I. & Maaten, L. v. d. *Self-supervised learning of pretext-invariant representations* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 6707–6717.

150. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. *Momentum contrast for unsupervised visual representation learning* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 9729–9738.

151. Giorgi, J., Nitski, O., Wang, B. & Bader, G. *DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations* in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Association for Computational Linguistics, Online, 2021), 879–895.

152. Tack, J., Mo, S., Jeong, J. & Shin, J. *CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances* in *34th Conference on Neural Information Processing Systems (NeurIPS) 2020* (2020).

153. Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., *et al.* Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566* (2020).

154. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., *et al.* Supervised Contrastive Learning. *Advances in Neural Information Processing Systems* **33** (2020).

155. Gunel, B., Du, J., Conneau, A. & Stoyanov, V. *Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning* in *International Conference for Learning Representations* (2021).

156. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., *et al. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank* in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Seattle, Washington, USA, 2013), 1631–1642.

157. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. *Learning Word Vectors for Sentiment Analysis* in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, Portland, Oregon, USA, 2011), 142–150.

158. Lang, K. in *Machine Learning Proceedings 1995* 331–339 (Elsevier, 1995).

159. Li, X. & Roth, D. *Learning question classifiers* in *COLING 2002: The 19th International Conference on Computational Linguistics* (2002).

160. Dagan, I., Glickman, O. & Magnini, B. *The PASCAL Recognising Textual Entailment Challenge* in *MLCW* (2005).

161. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D. & Magnini, B. *The Second PASCAL Recognising Textual Entailment Challenge* in (2006).

162. Giampiccolo, D., Magnini, B., Dagan, I. & Dolan, W. *The Third PASCAL Recognizing Textual Entailment Challenge* in *ACL-PASCAL@ACL* (2007).

163. Bentivogli, L., Clark, P., Dagan, I. & Giampiccolo, D. *The Sixth PASCAL Recognizing Textual Entailment Challenge* in *TAC* (2009).

164. Williams, A., Nangia, N. & Bowman, S. *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference* in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (Association for Computational Linguistics, New Orleans, Louisiana, 2018), 1112–1122.

165. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., *et al. Findings of the 2016 conference on machine translation* in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers* (2016), 131–198.

166. Elliott, D., Frank, S., Sima'an, K. & Specia, L. *Multi30K: Multilingual English-German Image Descriptions* in *Proceedings of the 5th Workshop on Vision and Language* (Association for Computational Linguistics, Berlin, Germany, 2016), 70–74.

167. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., *et al. Transformers: State-of-the-Art Natural Language Processing* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Online, 2020), 38–45.

168. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9** (2008).

169. Zhou, W., Zhang, S., Poon, H. & Chen, M. *Context-faithful Prompting for Large Language Models* in *Findings of the Association for Computational Linguistics: EMNLP 2023* (eds Bouamor, H., Pino, J. & Bali, K.) (Association for Computational Linguistics, Singapore, 2023), 14544–14556. doi:10.18653/v1/2023.findings-emnlp.968.

170. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., *et al.* Language models are few-shot learners. *Advances in neural information processing systems* **33,** 1877–1901 (2020).

171. Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., *et al. Finetuned Language Models are Zero-Shot Learners* in *International Conference on Learning Representations* (2022).

172. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., *et al.* Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

173. Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., *et al.* Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

174. Joshi, M., Choi, E., Weld, D. & Zettlemoyer, L. *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension* in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Vancouver, Canada, 2017), 1601–1611. doi:10.18653/v1/P17-1147.

175. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., *et al.* Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* **7,** 452–466. doi:10.1162/tacl_a_00276 (2019).

176. Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., *et al.* Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457* (2018).

177. Mihaylov, T., Clark, P., Khot, T. & Sabharwal, A. *Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Brussels, Belgium, 2018), 2381–2391. doi:10.18653/v1/D18-1260.

178. Lin, S., Hilton, J. & Evans, O. *TruthfulQA: Measuring How Models Mimic Human Falsehoods* in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Dublin, Ireland, 2022), 3214–3252. doi:10.18653/v1/2022.acl-long.229.

179. Lazaridou, A., Kuncoro, A., Gribovskaya, E., Agrawal, D., Liska, A., Terzi, T., *et al.* Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems* **34,** 29348–29363 (2021).

180. Liska, A., Kocisky, T., Gribovskaya, E., Terzi, T., Sezener, E., Agrawal, D., *et al.* *Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models* in *International Conference on Machine Learning* (2022), 13604–13622.

181. Kasai, J., Sakaguchi, K., Takahashi, Y., Bras, R. L., Asai, A., Yu, X., *et al.* RealTime QA: What's the Answer Right Now? *arXiv preprint arXiv:2207.13332* (2022).

182. Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M. & Toutanova, K. *BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions* 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)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 2924–2936. doi:10.18653/v1/N19-1300.

183. Rajpurkar, P., Zhang, J., Lopyrev, K. & Liang, P. *SQuAD: 100,000+ Questions for Machine Comprehension of Text* in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Austin, Texas, 2016), 2383–2392. doi:10.18653/v1/D16-1264.

184. Li, D., Rawat, A. S., Zaheer, M., Wang, X., Lukasik, M., Veit, A., *et al.* Large Language Models with Controllable Working Memory. *arXiv preprint arXiv:2211.05110* (2022).

185. Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J., *et al. Prompting gpt-3 to be reliable* in *International Conference on Learning Representations* (2023).

186. Longpre, S., Perisetla, K., Chen, A., Ramesh, N., DuBois, C. & Singh, S. *Entity-Based Knowledge Conflicts in Question Answering* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 7052–7063. doi:10.18653/v1/2021.emnlp-main.565.

187. Rajpurkar, P., Jia, R. & Liang, P. *Know What You Don't Know: Unanswerable Questions for SQuAD* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Association for Computational Linguistics, Melbourne, Australia, 2018), 784–789. doi:10.18653/v1/P18-2124.

188. Gupta, M., Kulkarni, N., Chanda, R., Rayasam, A. & Lipton, Z. C. *AmazonQA: A Review-Based Question Answering Task* in *International Joint Conference on Artificial Intelligence* (2019).

189. Bjerva, J., Bhutani, N., Golshan, B., Tan, W.-C. & Augenstein, I. *SubjQA: A Dataset for Subjectivity and Review Comprehension* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 5480–5494. doi:10.18653/v1/2020.emnlp-main.442.

190. Stoica, G., Platanios, E. A. & Póczos, B. *Re-tacred: Addressing shortcomings of the tacred dataset* in *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (2021), 13843–13850.

191. Wang, Y., Chen, M., Zhou, W., Cai, Y., Liang, Y., Liu, D., *et al.* Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. *arXiv preprint arXiv:2205.03784* (2022).

192. Fang, T., Wang, Z., Zhou, W., Zhang, H., Song, Y. & Chen, M. Getting sick after seeing a doctor? diagnosing and mitigating knowledge conflicts in event temporal reasoning. *arXiv preprint arXiv:2305.14970* (2023).

193. Wang, Y., Hooi, B., Wang, F., Cai, Y., Liang, Y., Zhou, W., *et al. How Fragile is Relation Extraction under Entity Replacements?* in *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)* (eds Jiang, J., Reitter, D. & Deng, S.) (Association for Computational Linguistics, Singapore, 2023), 414–423. doi:10.18653/v1/2023.conll-1.27.

194. Wang, F., Mo, W., Wang, Y., Zhou, W. & Chen, M. *A Causal View of Entity Bias in (Large) Language Models* in *Findings of the Association for Computational Linguistics: EMNLP 2023* (eds Bouamor, H., Pino, J. & Bali, K.) (Association for Computational Linguistics, Singapore, 2023), 15173–15184. doi:10.18653/v1/2023.findings-emnlp.1013.

195. Zhu, C., Rawat, A. S., Zaheer, M., Bhojanapalli, S., Li, D., Yu, F., *et al.* Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363* (2020).

196. De Cao, N., Aziz, W. & Titov, I. *Editing Factual Knowledge in Language Models* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 6491–6506. doi:10.18653/v1/2021.emnlp-main.522.

197. Mitchell, E., Lin, C., Bosselut, A., Finn, C. & Manning, C. D. *Fast Model Editing at Scale* in *International Conference on Learning Representations* (2022).

198. Meng, K., Bau, D., Andonian, A. J. & Belinkov, Y. *Locating and editing factual associations in gpt* in *Advances in Neural Information Processing Systems* (2022).

199. Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y. & Bau, D. *Mass-editing memory in a transformer* in *International Conference on Learning Representations* (2023).

200. Lazaridou, A., Gribovskaya, E., Stokowiec, W. & Grigorev, N. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115* (2022).

201. Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., *et al.* Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* (2022).

202. Khattab, O., Santhanam, K., Li, X. L., Hall, D., Liang, P., Potts, C., *et al.* Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024* (2022).

203. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., *et al. Dense Passage Retrieval for Open-Domain Question Answering* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

204. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C. & Zaharia, M. *ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction* in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, Seattle, United States, 2022), 3715–3734. doi:10.18653/v1/2022.naacl-main.272.

205. Gao, L. & Callan, J. *Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval* in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Dublin, Ireland, 2022), 2843–2853. doi:10.18653/v1/2022.acl-long.203.

206. Neeman, E., Aharoni, R., Honovich, O., Choshen, L., Szpektor, I. & Abend, O. DisentQA: Disentangling Parametric and Contextual Knowledge with Counterfactual Question Answering. *arXiv preprint arXiv:2211.05655* (2022).

207. Chow, C. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory* **16,** 41–46 (1970).

208. Fumera, G. & Roli, F. *Support vector machines with embedded reject option* in *Pattern Recognition with Support Vector Machines: First International Workshop, SVM 2002 Niagara Falls, Canada, August 10, 2002 Proceedings* (2002), 68–82.

209. Cortes, C., DeSalvo, G. & Mohri, M. *Learning with rejection* in *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27* (2016), 67–82.

210. Wang, F., Huang, J. Y., Yan, T., Zhou, W. & Chen, M. *Robust Natural Language Understanding with Residual Attention Debiasing* in *Findings of the Association for Computational Linguistics: ACL 2023* (eds Rogers, A., Boyd-Graber, J. & Okazaki, N.) (Association for Computational Linguistics, Toronto, Canada, 2023), 504–519. doi:10.18653/v1/2023.findings-acl.32.

211. Gal, Y. & Ghahramani, Z. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning* in *international conference on machine learning* (2016), 1050–1059.

212. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* **30** (2017).

213. Xin, J., Tang, R., Yu, Y. & Lin, J. *The Art of Abstention: Selective Prediction and Error Regularization for Natural Language Processing* in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Association for Computational Linguistics, Online, 2021), 1040–1051. doi:10.18653/v1/2021.acl-long.84.

214. Yatskar, M. *A Qualitative Comparison of CoQA, SQuAD 2.0 and QuAC* 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)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 2318–2323. doi:10.18653/v1/N19-1241.

215. Reddy, S., Chen, D. & Manning, C. D. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* **7,** 249–266. doi:10.1162/tacl_a_00266 (2019).

216. Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., *et al. QuAC: Question Answering in Context* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Brussels, Belgium, 2018), 2174–2184. doi:10.18653/v1/D18-1241.

217. Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., *et al.* Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910* (2022).

218. Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L. & Chen, W. *What Makes Good In-Context Examples for GPT-3?* in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures* (Association for Computational Linguistics, Dublin, Ireland and Online, 2022), 100–114. doi:10.18653/v1/2022.deelio-1.10.

219. Reimers, N. & Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 3982–3992. doi:10.18653/v1/D19-1410.

220. Zeng, D., Liu, K., Chen, Y. & Zhao, J. *Distant supervision for relation extraction via piecewise convolutional neural networks* in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), 1753–1762.

221. Mintz, M., Bills, S., Snow, R. & Jurafsky, D. *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: Volume 2-Volume 2* (2009), 1003–1011.

222. Surdeanu, M., Tibshirani, J., Nallapati, R. & Manning, C. D. *Multi-instance multi-label learning for relation extraction* in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (2012), 455–465.

223. Nakashole, N., Weikum, G. & Suchanek, F. *PATTY: a taxonomy of relational patterns with semantic types* in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2012), 1135–1145.

224. Jiang, M., Shang, J., Cassidy, T., Ren, X., Kaplan, L. M., Hanratty, T. P., *et al. Metapad: Meta pattern discovery from massive text corpora* in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), 877–886.

225. Jones, R., McCallum, A., Nigam, K. & Riloff, E. *Bootstrapping for text learning tasks* in *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications* **1** (1999).

226. Agichtein, E. & Gravano, L. *Snowball: Extracting relations from large plain-text collections* in *Proceedings of the fifth ACM conference on Digital libraries* (2000), 85–94.

227. Batista, D. S., Martins, B. & Silva, M. J. *Semi-supervised bootstrapping of relationship extractors with distributional semantics* in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), 499–504.

228. Hancock, B., Varma, P., Wang, S., Bringmann, M., Liang, P. & Ré, C. Training Classifiers with Natural Language Explanations. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* doi:10.18653/v1/p18-1175 (2018).

229. Rosenberg, C., Hebert, M. & Schneiderman, H. Semi-supervised self-training of object detection models (2005).

230. Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., *et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals* in *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions* (2009), 94–99.

231. Lin, H., Yan, J., Qu, M. & Ren, X. *Learning Dual Retrieval Module for Semi-supervised Relation Extraction* in *The Web Conference* (2019).

232. Roth, B., Barth, T., Wiegand, M., Singh, M. & Klakow, D. Effective slot filling based on shallow distant supervision methods. *arXiv preprint arXiv:1401.1158* (2014).

233. Willett, P. The Porter stemming algorithm: then and now. *Program* **40,** 219–223 (2006).

234. Qu, M., Ren, X., Zhang, Y. & Han, J. *Weakly-supervised Relation Extraction by Pattern-enhanced Embedding Learning* in *Proceedings of the 2018 World Wide Web Conference on World Wide Web* (2018), 1257–1266.

235. Wang, L., Cao, Z., de Melo, G. & Liu, Z. *Relation Classification via Multi-Level Attention CNNs* in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Berlin, Germany, 2016), 1298–1307. doi:10.18653/v1/P16-1123.

236. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., *et al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification* in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Association for Computational Linguistics, Berlin, Germany, 2016), 207–212. doi:10.18653/v1/P16-2034.

237. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

238. Yu, M. & Dredze, M. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics* **3,** 227–242 (2015).

239. Neculoiu, P., Versteegh, M. & Rotaru, M. *Learning text similarity with siamese recurrent networks* in *Proceedings of the 1st Workshop on Representation Learning for NLP* (2016), 148–157.

240. Lee, D.-H. *Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks* in (2013).

241. Jiang, J. & Zhai, C. *Instance weighting for domain adaptation in NLP* in *Proceedings of the 45th annual meeting of the association of computational linguistics* (2007), 264–271.

242. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. *Distributed representations of words and phrases and their compositionality* in *Advances in neural information processing systems* (2013), 3111–3119.

243. Pennington, J., Socher, R. & Manning, C. *Glove: Global vectors for word representation* in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), 1532–1543.

244. Li, S., Xu, H. & Lu, Z. Generalize Symbolic Knowledge With Neural Rule Engine. *arXiv preprint arXiv:1808.10326* (2018).

245. Tarvainen, A. & Valpola, H. *Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results* in *Advances in neural information processing systems* (2017), 1195–1204.

246. Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, *et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* Software available from tensorflow.org. 2015.

247. Duchi, J., Hazan, E. & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* **12,** 2121–2159 (2011).

248. Hearst, M. A. *Automatic acquisition of hyponyms from large text corpora* in *Proceedings of the 14th conference on Computational linguistics-Volume 2* (1992), 539–545.

249. Lu, K., Hsu, I.-H., Zhou, W., Ma, M. D. & Chen, M. *Multi-hop Evidence Retrieval for Cross-document Relation Extraction* in *Findings of the Association for Computational Linguistics: ACL 2023* (eds Rogers, A., Boyd-Graber, J. & Okazaki, N.) (Association for Computational Linguistics, Toronto, Canada, 2023), 10336–10351. doi:10.18653/v1/2023.findings-acl.657.

250. Srivastava, S., Labutov, I. & Mitchell, T. M. *Joint Concept Learning and Semantic Parsing from Natural Language Explanations* in *EMNLP* (2017).

251. Xu, W., Sun, H., Deng, C. & Tan, Y. *Variational autoencoder for semi-supervised text classification* in *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

252. Zeng, Z., Zhou, W., Liu, X. & Song, Y. A variational approach to weakly supervised document-level multi-aspect sentiment classification. *arXiv preprint arXiv:1904.05055* (2019).

253. Gupta, P., Roth, B. & Schütze, H. Joint bootstrapping machines for high confidence relation extraction. *arXiv preprint arXiv:1805.00254* (2018).

254. Zhou, W., Zhang, S., Naumann, T., Chen, M. & Poon, H. *Continual Contrastive Finetuning Improves Low-Resource Relation Extraction* in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (eds Rogers, A., Boyd-Graber, J. & Okazaki, N.) (Association for Computational Linguistics, Toronto, Canada, 2023), 13249–13263. doi:10.18653/v1/2023.acl-long.739.

255. Nan, G., Guo, Z., Sekulic, I. & Lu, W. *Reasoning with Latent Structure Refinement for Document-Level Relation Extraction* in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Online, 2020), 1546–1557. doi:10.18653/v1/2020.acl-main.141.

256. Baldini Soares, L., FitzGerald, N., Ling, J. & Kwiatkowski, T. *Matching the Blanks: Distributional Similarity for Relation Learning* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, 2019), 2895–2905. doi:10.18653/v1/P19-1279.

257. Peng, H., Gao, T., Han, X., Lin, Y., Li, P., Liu, Z., *et al. Learning from Context or Names? An Empirical Study on Neural Relation Extraction* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 3661–3672. doi:10.18653/v1/2020.emnlp-main.298.

258. Qin, Y., Lin, Y., Takanobu, R., Liu, Z., Li, P., Ji, H., *et al. ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning* in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Association for Computational Linguistics, Online, 2021), 3350–3363. doi:10.18653/v1/2021.acl-long.260.

259. Mintz, M., Bills, S., Snow, R. & Jurafsky, D. *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* (Association for Computational Linguistics, Suntec, Singapore, 2009), 1003–1011.

260. Hadsell, R., Chopra, S. & LeCun, Y. *Dimensionality reduction by learning an invariant mapping* in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* **2** (2006), 1735–1742.

261. Graf, F., Hofer, C., Niethammer, M. & Kwitt, R. *Dissecting supervised constrastive learning* in *International Conference on Machine Learning* (2021), 3821–3830.

262. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L. & Lewis, M. *Generalization through Memorization: Nearest Neighbor Language Models* in *International Conference on Learning Representations* (2020).

263. Khandelwal, U., Fan, A., Jurafsky, D., Zettlemoyer, L. & Lewis, M. *Nearest Neighbor Machine Translation* in *International Conference on Learning Representations* (2021).

264. Jia, R., Wong, C. & Poon, H. *Document-Level N-ary Relation Extraction with Multiscale Representation Learning* 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)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 3693–3704. doi:10.18653/v1/N19-1370.

265. Tan, Q., Xu, L., Bing, L. & Ng, H. T. Revisiting DocRED - Addressing the Overlooked False Negative Problem in Relation Extraction. *arXiv preprint arXiv:2205.12696* (2022).

266. Quirk, C. & Poon, H. *Distant Supervision for Relation Extraction beyond the Sentence Boundary* in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (Association for Computational Linguistics, Valencia, Spain, 2017), 1171–1182.

267. Peng, N., Poon, H., Quirk, C., Toutanova, K. & Yih, W.-t. Cross-Sentence N-ary Relation Extraction with Graph LSTMs. *Transactions of the Association for Computational Linguistics* **5,** 101–115. doi:10.1162/tacl_a_00049 (2017).

268. Christopoulou, F., Miwa, M. & Ananiadou, S. *Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Association for Computational Linguistics, Hong Kong, China, 2019), 4925–4936. doi:10.18653/v1/D19-1498.

269. Zeng, S., Xu, R., Chang, B. & Li, L. *Double Graph Based Reasoning for Document-level Relation Extraction* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, Online, 2020), 1630–1640. doi:10.18653/v1/2020.emnlp-main.127.

270. Zhang, N., Chen, X., Xie, X., Deng, S., Tan, C., Chen, M., *et al. Document-level Relation Extraction as Semantic Segmentation* in *IJCAI* (2021).

271. Tan, Q., He, R., Bing, L. & Ng, H. T. *Document-Level Relation Extraction with Adaptive Focal Loss and Knowledge Distillation* in *Findings of the Association for Computational Linguistics: ACL 2022* (Association for Computational Linguistics, Dublin, Ireland, 2022), 1672–1681. doi:10.18653/v1/2022.findings-acl.132.

272. Ronneberger, O., Fischer, P. & Brox, T. *U-net: Convolutional networks for biomedical image segmentation* in *International Conference on Medical image computing and computer-assisted intervention* (2015), 234–241.

273. Levy, O., Seo, M., Choi, E. & Zettlemoyer, L. *Zero-Shot Relation Extraction via Reading Comprehension* in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)* (Association for Computational Linguistics, Vancouver, Canada, 2017), 333–342. doi:10.18653/v1/K17-1034.

274. Lu, K., Hsu, I.-H., Ma, M. D., Zhou, W. & Chen, M. *Summarization as Indirect Supervision for Relation Extraction* in *Findings of ACL: EMNLP* (2022).

275. Zhou, W., Liu, F., Vulić, I., Collier, N. & Chen, M. Prix-lm: Pretraining for multilingual knowledge base construction. *arXiv preprint arXiv:2110.08443* (2021).

276. Li, Z., Zhou, W., Chiang, Y.-Y. & Chen, M. Geolm: Empowering language models for geospatially grounded language understanding. *arXiv preprint arXiv:2310.14478* (2023).

277. Gao, T., Yao, X. & Chen, D. *SimCSE: Simple Contrastive Learning of Sentence Embeddings* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 6894–6910. doi:10.18653/v1/2021.emnlp-main.552.

278. Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., *et al.* Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)* **3,** 1–23 (2021).

279. Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S. & Singh, S. *No fuss distance metric learning using proxies* in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 360–368.

280. Zhu, J., Wang, Z., Chen, J., Chen, Y.-P. P. & Jiang, Y.-G. *Balanced Contrastive Learning for Long-Tailed Visual Recognition* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 6908–6917.

281. Christobel, Y. A. & Sivaprakasam, P. A new classwise k nearest neighbor (CKNN) method for the classification of diabetes dataset. *International Journal of Engineering and Advanced Technology* **2,** 396–200 (2013).

282. Zhang, N., Deng, S., Sun, Z., Wang, G., Chen, X., Zhang, W., *et al. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks* 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)* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019), 3016–3025. doi:10.18653/v1/N19-1306.

283. Wei, C.-H., Allot, A., Leaman, R. & Lu, Z. PubTator central: automated concept annotation for biomedical full text articles. *Nucleic acids research* **47,** W587–W593 (2019).

284. Loshchilov, I. & Hutter, F. *Decoupled Weight Decay Regularization* in *International Conference on Learning Representations* (2018).

285. Islam, A., Chen, C.-F. R., Panda, R., Karlinsky, L., Radke, R. & Feris, R. *A broad study on the transferability of visual representations with contrastive learning* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 8845–8855.

286. Chen, M., Fu, D. Y., Narayan, A., Zhang, M., Song, Z., Fatahalian, K., *et al. Perfectly balanced: Improving transfer and robustness of supervised contrastive learning* in *International Conference on Machine Learning* (2022), 3090–3122.

287. Wang, F. & Liu, H. *Understanding the behaviour of contrastive loss* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), 2495–2504.

288. McClosky, D., Surdeanu, M. & Manning, C. *Event Extraction as Dependency Parsing* in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, Portland, Oregon, USA, 2011), 1626–1635.

289. Yao, Y., Du, J., Lin, Y., Li, P., Liu, Z., Zhou, J., *et al. CodRED: A Cross-Document Relation Extraction Dataset for Acquiring Knowledge in the Wild* in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), 4452–4472. doi:10.18653/v1/2021.emnlp-main.366.

290. Borji, A. A categorical archive of ChatGPT failures. *arXiv preprint arXiv:2302.03494* (2023).