Pre-trained Language Models as Re-Annotators

Chang Shu

Master of Science by Research
School of Philosophy, Psychology and Language Sciences
University of Edinburgh
2022

Abstract

Annotation noise are widespread in datasets, but manually revising a flawed corpus is time-consuming and error-prone. Hence, given the prior knowledge in Pre-trained Language Models and the expected uniformity across all annotations, we attempt to reduce annotation noise in the corpus through two tasks automatically: (1) Annotation Inconsistency Detection that indicates the credibility of annotations, and (2) Annotation Error Correction that rectifies the abnormal annotations.

We investigate how to acquire semantic sensitive annotation representations from Pre-trained Language Models, expecting to embed the examples with identical annotations to the mutually adjacent positions even without fine-tuning. We proposed a novel credibility score to reveal the likelihood of annotation inconsistencies based on the neighbouring consistency. Then, we fine-tune the Pre-trained Language Models based classifier with cross-validation for annotation correction. The annotation corrector is further elaborated with two approaches: (1) soft labelling by Kernel Density Estimation and (2) a novel distant-peer contrastive loss.

We study the re-annotation in relation extraction and create a new manually revised dataset, Re-DocRED, for evaluating document-level re-annotation. The proposed credibility scores show promising agreement with human revisions, achieving a Binary F_1 of 93.4 and 72.5 in detecting inconsistencies on TACRED and DocRED respectively. Moreover, the neighbour-aware classifiers based on distant-peer contrastive learning and uncertain labels achieve Macro F_1 up to 66.2 and 57.8 in correcting annotations on TACRED and DocRED respectively. These improvements are not merely theoretical: Rather, automatically denoised training sets demonstrate up to 3.6% performance improvement for state-of-the-art relation extraction models, and the proposed framework is expected to be hundreds of times faster than the human re-annotators empirically.

Acknowledgements

This dissertation is dedicated to my father, Shengguo Shu. I wish you a happy birthday, and thank you for always being there.

I am deeply grateful to my supervisors, Prof. Bonnie Webber, Dr. Beatrice Alex and Andreas Grivas, for bringing me to this exciting project and continuous support. I believe they are some of the best supervisors and NLP researchers on the planet, and it is such a great honour to work with them. Additionally, I would like to thank my personal tutor, Dr. Catherine Lai, for her help and advice during my master study. I also appreciate Luxi He for proofreading and Anda Zhou for discussion through this dissertation.

I would also like to express gratitude to my previous supervisors, Dr. Rui Zhang, Dr. Tao Yu, Dr. Jian Qiu, and Prof. Zhiyuan Liu, for their past instructions and kindness in offering internship opportunities at Penn State University, Yale University, Alibaba Cloud and Tsinghua University. I also appreciate all my mentors and colleagues during these internships, Peng Shi, Jie Zhou, Taiyan Li, Yusen Zhang and Xiangyu Dong.

Finally, I would like to say my deepest thanks to my kith and kin. It was a miserable year for me physically and mentally, and I definitely would not make it without your endless support and love.

The Tao that can be told is not the eternal Tao. Though findings in this dissertation are ephemeral, I am thankful for the undiluted pleasure brought by this exploration.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Chang Shu*)

Table of Contents

Chapter 1

Introduction

Annotation noise is pervasive in datasets and becomes increasingly problematic as data-driven methods are increasingly incorporated into Natural Language Processing (NLP). This dissertation is the first to leverage prior knowledge in Pre-trained Language Models to detect annotation noise and inconsistencies and correct annotation errors. We introduce the prime motivation behind this project and outline the investigations we conducted to approach this problem. We also summarise our key contributions and the main contents of each chapter in the dissertation.

1.1 Motivation

Recent decades have witnessed profound shifts in NLP research from symbolic methods to statistical techniques, and then to neural methods (Khurana et al., 2017). Early research in NLP mainly relied on a finite set of hand-written rules that reflected common linguistic knowledge. In contrast, the latest paradigm of NLP involves letting neural models learn latent linguistic knowledge from vast amounts of data. As data-driven methods dominate NLP research, the importance of annotation quality is increasingly visible. However, most of the recent advances in NLP still focus on developing models with enhanced representation learning capability, underestimating the deterioration of model performance caused by annotation noise in training data (Larson et al., 2020; Khayrallah and Koehn, 2018) and misdirection of model evaluation caused by the flawed test data (Northcutt et al., 2021a). The main reason behind the phenomenon is that annotating a dataset is time-consuming, high-cost and labour-intensive, as is revising noisy and/or inconsistent datasets. Hence, an automatic re-annotator that can partially reduce labour costs or even fully replace human labour will be of benefit to the field of NLP.

The surge of Pre-trained Language Models (PLMs) is one of the most significant revolutions that has emerged in the era of neural NLP (Qiu et al., 2020; Min et al., 2021). Instead of domain-specific learning, PLMs are first unsupervised, pre-trained on the large-scale corpus, and then fine-tuned on downstream tasks. Recent studies suggest that the pre-training stage endows the PLMs with abundant commonsense (Peters et al., 2019; Davison et al., 2019; Jiang et al., 2020a,b) and linguistic knowledge (Clark et al., 2019,?; Liu et al., 2019a; Chi et al., 2020; Ettinger, 2020). The prior knowledge in PLMs has proved to be versatile in practice. For instance, PLMs can be directly used to evaluate text generation (Zhang et al., 2020; Sellam et al., 2020) and probe factual knowledge (Peters et al., 2019). Considering the appealing property of PLMs, we are curious whether they can contribute to automatic re-annotation.

The prime motivation and novelty of this project involves applying Pre-trained Language Models as re-annotators to improve annotation quality with reduced cost and competitive results. As argued by Dickinson and Meurers (2003), examples that have different labels but occur in very similar contexts are likely to be annotation inconsistencies or errors. Coincidentally, PLMs are well known for their outstanding capability of acquiring contextualized embedding. Therefore, the main idea of detecting or correcting diverging annotations with PLMs is to contrast the label of the target example with other examples embedded in its vicinity.

1.2 Investigations

We are the first to comprehensively study the potential of PLMs in data re-annotation using both a sentence-level and a document-level relation extraction dataset, TACRED (Zhang et al., 2017a) and DocRED (Yao et al., 2019b). Relation extraction is the task of determining the relation holding between two entities in context – one called the *subject*, the other, the *object*. We are evaluating re-annotators for sentence-level relation extraction using two datasets derived from TACRED — TACRev (Alt et al., 2020) and Re-TACRED (Stoica et al., 2021) with human revisions. To evaluate the re-annotators in document-level relation extraction, we re-annotated a subset of DocRED ourselves, to create a novel human revised dataset annotated with document-level relations.

The re-annotation task is composed of two steps: **Annotation Inconsistency Detection** and **Annotation Error Correction** (Figure 1.1). Annotation Inconsistency Detection (AID) evaluates the consistency of each given annotation compared to other annotations in a similar context. Annotation Error Correction (AEC) suggests the proper annotation for the example identified as an anomaly.

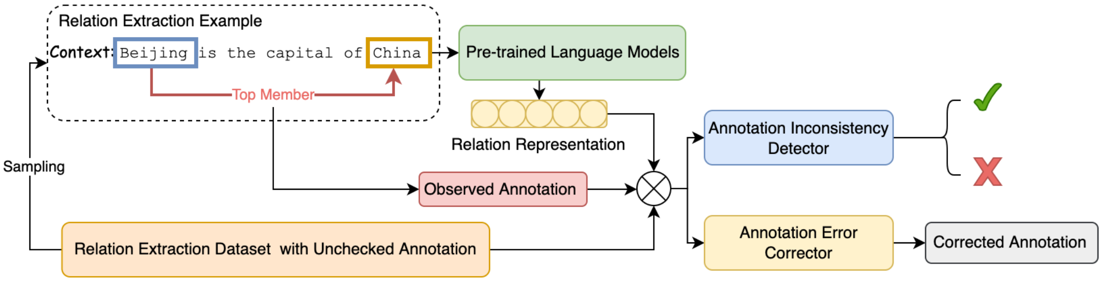

Relation Extraction Example
Context: Beijing is the capital of China
Top Member
Pre-trained Language Models
Relation Representation
Sampling
Observed Annotation
Annotation Inconsistency Detector
✓
✗
Annotation Error Corrector
Corrected Annotation
Relation Extraction Dataset with Unchecked Annotation

Figure 1.1: The overview of our proposed Annotation Inconsistency Detector (AID) and Annotation Error Corrector (AEC) in the context of relation extraction. The **Annotation Inconsistency Detector** indicates whether the observed annotation is consistent with other annotations. The **Annotation Error Corrector** rectifies the annotations identified as mislabeled.

Annotation Inconsistency Detection relies on the favourable property of representation methods and the desired sensitivity and specificity of the noise detector. Instead of fine-tuning the PLMs, we first researched different prompt (Liu et al., 2021a) and input modification (Zhou and Chen, 2021) techniques to acquire informative and distinguishable relation representation of each instance from PLMs. After optimizing the relation embedder, we leverage the K-Nearest Neighbour algorithm (Mucherino et al., 2009) to determine potentially inconsistent annotations based on the local geometry of annotated examples in the embedding space. Furthermore, we propose a novel distance-based credibility score that jointly considers the labels in the vicinity and the global distribution of the assigned class of the query example. The experiments reveal that inadequately informative and overly predisposed prompts result in downgraded performance in detecting inconsistency. Empirically, our proposed credibility scores combined with relation prompts show promising agreement with human revisions in Re-TACRED and TACRev, reaching a binary F_1 score of 93.4 and 72.5 in detecting inconsistencies on TACRED and DocRED respectively.

On the other hand, the PLM-based Annotation Error Corrector is fine-tuned by cross-validation Stone (1977); Tibshirani (1996); Allen (1974) to make accurate revision decisions. The vanilla automatic corrector comprises PLMs stacked by a neural relation classifier. We ameliorate the learning process of AEC models with uncertain labels and neighbour-aware learning. Inspired by soft labels (Thiel, 2008; Nguyen et al., 2014; Liu et al., 2017; Zhao et al., 2014; Algan and Ulusoy, 2021), we construct the labels with uncertainty of samples based on their neighbours or estimated probability densities regarding each label class to replace overconfident hard labels. Specifically, one approach replaces part of hard labels with the majority labels among their neighbours, and another derives the soft-label vectors from the estimated kernel density corresponding to every class. We also explore two methods to augment the relation classifier with neighbouring knowledge: (1) rank-aware Transformer encoder (Vaswani et al., 2017) to acquire the relation embedding attended on their neighbours, and (2) distant-peer contrastive learning (Khosla et al., 2020) to include neighbour information to the loss function. Apart from sampling the positives and negatives in the batch, the framework of distant-peer contrastive learning selects positive examples from neighbours by our proposed peer distance computed by combining their cooccurrence frequency and squared Euclidean distance. Our empirical results show that the neighbour-aware annotation corrector trained with distant-peer contrastive learning obtains macro F_1 up to 66.2 on TACRED and 57.8 on DocRED. Moreover, training state-of-the-art relation extraction models on the training sets automatically denoised by our optimized annotation corrector leads to the maximum improvement of micro F_1 of 3.5% on TACRED, 3.4% on TACRev, and 1.1% on DocRED.

Finally, we found that our proposed Annotation Error Corrector could automatically revise the annotations hundreds of times faster than human revisers with acceptable reliability. Similarly, an annotation inconsistency detector could spot dubious annotations even over ten thousand times faster than humans. Therefore, we believe that applying automatic re-annotators prior to manual revisions or even entirely relying on their revising outcomes would considerably improve the quality of data and data-driven NLP.

1.3 Contributions

The main contributions of the dissertation are:

- **RE-DocRED Dataset**: To study the annotation quality and evaluate our proposed automatic re-annotator in document-level tasks, we built a novel dataset, Re-DocRED, by revising 411 examples in the document-level relation extraction dataset, DocRED. It is the first re-annotated RE dataset at the document-level and will benefit future research on annotation noise.

- **PLMs for Re-Annotation**: We are the first to leverage prior knowledge in Pre-trained Language Models to detect annotation inconsistencies and correct annotation errors. Our findings prove that factual and linguistic expertise in Pre-trained Language Models is applicable to automatic re-annotation even in the zero-shot scenario.

- **Prompt Investigation**: We comprehensively study the impact of different forms of prompts and input modifications to the re-annotation tasks. Empirically, we found that prompts with either inadequate contexts or strong implications would mislead automatic re-annotators.
- **Credibility Score**: We propose a novel credibility score jointly computed by the distance and reliability of neighbours. The reliability of neighbours is approximated by the estimated kernel density of their assigned classes. The experiment indicates the credibility score is very effective in spotting potential inconsistent annotations.
- **Neighbour-based Uncertain Label**: We construct the labels with uncertainty based on the distribution of neighbours to avoid the annotation corrector from relying overly on the observed hard labels. Both the K-Nearest Neighbours based label replacements and Kernel Density Estimation based soft labels show convincing improvement to the corrector performance.
- **Distant-peer Contrastive Learning**: We develop a novel contrastive learning framework with augmented positive examples chosen by our newly defined peer distance. Based on their co-occurrence frequency and distance, we formulate the peer distance to select the most trustful and valuable positives from the neighbours of query examples. The results demonstrate the strength of combining distant-peer contrastive loss and the cross-entropy loss, especially when used in collaboration with with Kernel Density Estimation based soft labels.

1.4 Dissertation Structure

The summaries of the following chapters of the dissertation are listed as follows:

- **Chapter 2** introduces the previous work in (1) prior knowledge in Pre-trained Language Models and prompts for its transferability, (2) analysis of annotation noise and noise-tolerant learning methods, and (3) efforts in improving annotation quality manually and automatically.

- **Chapter 3** defines the two investigated tasks, Annotation Inconsistency Detection and Annotation Error Correction in the context of relation extraction, and describe the datasets for experiments.

- **Chapter 4** presents our empirical study in relation representation methods and neighbouring consistency based binary classifiers for Annotation Inconsistency Detection.

- **Chapter 5** describes our investigation in cross-validation based Annotation Error Correction, and two improvements, uncertain labels and neighbouring awareness.

- **Chapter 6** concludes our findings throughout the study and presents several charming directions to be explored in future.

Chapter 2

Related Work

The three topics of most importance to this dissertation are: (1) Pre-trained Language Models; (2) identifying noise and inconsistencies that can arise in annotation, focusing on the annotation of relations rather than just on the annotation of simple strings; and (3) improving the annotation of relations through automating attempts to recognize and correct noisy or inconsistent tokens. We will address previous work on each of these topics in its own subsection, in order to clarify and justify the work we have done here.

2.1 Pre-trained Language Models

Various Natural Language Processing tasks focus on an independent domain, but some tasks are intrinsically connected. For instance, while Dependency Parsing (Dozat and Manning, 2017; Kübler et al., 2009; Nivre, 2005; Li et al., 2018) generates syntactic dependency trees, and while Semantic Role Labeling (SRL) (Palmer et al., 2010; Màrquez et al., 2008) predicts the latent predicate-argument structure, both tasks require the models to have the fundamental capability of capturing the grammatical structure of the given context. Hence, the paradigm in NLP recently shifted from task-specific model designs to a pre-train and fine-tuned pipeline.

Typically, a sequence-to-sequence model with abundant parameters is trained on a massive corpus to perform language modelling tasks or text reconstruction tasks in an unsupervised learning fashion during the pre-training stage and then the model parameters are fine-tuned subtly with the task-specific data and learning objectives such as relation extraction and sentiment analysis. On the basis of its characteristics, those models are widely known as the Pre-trained Language Models (PLMs). According to the intended usage scenario, the PLMs can be roughly divided into general-purpose

PLMs and special-purpose PLMs. The general-purpose PLMs are usually pre-trained on the general corpus, such as Wikipedia, and with the fundamental pre-training tasks and model architecture (Devlin et al., 2019; Joshi et al., 2020; Brown et al., 2020; Raffel et al., 2020; Lewis et al., 2020; Yang et al., 2019; Liu et al., 2019b). Those PLMs are more widely used and performed evenly on diverse downstream NLP tasks. Still, for those tasks demanding professional knowledge, they may be incapable of performing effectively because of lacking domain-specific pre-training. Therefore, the special-purpose PLMs normally pre-trained on a professional corpus (Lee et al., 2020; Feng et al., 2020; Li et al., 2020b), either refine the common model architecture of a general-purpose model (Peters et al., 2019; Zhang et al., 2019) or add auxiliary pre-training tasks (Soares et al., 2019).

Our target is to assist in mitigating the annotation noise and improve the quality of datasets. This dissertation is the first to comprehensively investigate the possible roles that PLMs can play in automatic re-annotation. To revise any existing annotations presented in the datasets, one needs to have two kinds of prior knowledge: (1) Factual and linguistic knowledge for revising the annotations that do not comply with common sense; (2) Knowledge of overall annotation distributions for detecting annotations that appear inconsistent with most other annotations. Recent studies reveal that most PLMs actually possess ample general factual and linguistic knowledge even without any fine-tuning or knowledge injection, such as Peters et al. (2019) and Hewitt and Manning (2019b). Therefore, we focus on developing the framework for automatic re-annotation based on the general-purpose PLMs. The next section (Section 2.1.1) discusses the recent advances in probing the prior knowledge of general-purpose PLMs. Section 2.1.2 then discusses the popular prompt-based learning for deriving the knowledge of interests from PLMs and applying it to the downstream tasks. Both directions motivate the methodologies we adopt to acquire the information-rich and highly differential representations of the examples in the dataset for automatic reexamination.

2.1.1 Prior Knowledge in Pre-trained Language Models

Just as bookworms can become encyclopedic by large amounts of reading, PLMs are also likely to acquire extensive linguistic and commonsense knowledge via unsupervised learning on the large-scale general corpus. Many empirical and analytical investigations have been conducted recently to justify this intuition for probing and quantifying the underlying prior knowledge in PLMs. Those findings doubtlessly breed sufficient confidence of trusting and leveraging prior knowledge in PLMs to revise the existing labels in flawed datasets.

Factual probing uncovers the hidden knowledge from PLMs firstly proposed by Peters et al. (2019). To explore the solutions for this task, they introduce the LAMA (LAnguage Model Analysis) framework, which is intended to probe the commonsense knowledge in PLMs. According to the proposed framework, the entries in the knowledge bases are converted into cloze statements with templates where the relation or entity mentions are replaced with the mask. Then the prior knowledge in PLMs are assessed by their performance in completing the missing tokens. For instance, given the cloze statement ”Newton was born in [MASK]”, if the PLMs are able to rank the ”UK” or ”England” higher for filling the blank, they would be regarded as having more factual knowledge. Their experimental results convince that PLMs without fine-tuning still contain trustworthy relational knowledge and can handle open-domain questions based on their factual knowledge. Davison et al. (2019) also develop a similar framework for mining the commonsense knowledge from PLMs. They first derive the masked sentences from the relational triples and then leverage the PLMs to rank the validity of a triple with the estimated point-wise mutual information between two entity mentions without fine-tuning. X-FACTOR by Jiang et al. (2020a) attempt to generalize the cloze-style factual probing to a multi-lingual situation by composing the variations of cloze templates in 23 typologically divergent languages and proposing an improvement for probing multilingual knowledge from PLMs based on code-switching. Their experiments demonstrate the multilingual accessibility of factual knowledge in PLM. The work by Jiang et al. (2020b) further optimizes cloze-base querying processing for more accurate estimation of factual knowledge in PLMs by replacing the manually crafted prompts with an automatic pipeline that generates templates based on the paraphrasing and intention mining. They suggest that the quality or compatibility of the prompts could impact the performance of knowledge probers of PLMs.

Aside from the commonsense knowledge, PLMs are to be able to comprehend a considerable amount of linguistic phenomena by merely pre-training on large-scale plain text without explicit linguistic annotations. Hewitt and Manning (2019b) testify syntactic knowledge in PLMs by showing that syntax trees are consistently embedded in the representation space of PLMs following certain liner transformations. Clark et al. (2019) also confirm the existence of syntactic knowledge in PLMs through the empirical analysis of the attention distribution, which displays the pattern of directing objects of prepositions and verbs, determiners of nouns or co-referent mentions. Tenney et al. (2019) indicates that PLMs have the latent procedure of handling linguistic information similar to a conventional NLP pipeline of part-of-speech tagging, dependency parsing, named entity recognition, and then co-reference. Through sixteen various probing tasks, Liu et al. (2019a) investigate the detailed impacts of pre-training tasks on the linguistic knowledge learned by PLMs and prove that PLMs have the knowledge of semantic dependency and co-reference resolution. Based on the analysis of word embedding space by multilingual PLMs, Chi et al. (2020) discover the universal grammatical relations across languages captured by PLMs. Ettinger (2020) propose a novel suite of diagnostics derived from human language experiments to examine the linguistic knowledge in PLMs. They suggest that PLMs can robustly identify good from bad completions involving shared category or role reversal and retrieve noun hypernyms but are slightly hesitant compared to the human evaluators.

2.1.2 Prompts for Knowledge Transferability

Prompt-based learning, an straightforward method of applying the prior knowledge in PLMs to the downstream tasks, has drawn more attention recently because it supports zero-shot learning. Instead of time-consuming adaptation of all parameters in PLMs according to the downstream learning objective, prompt-based learning reformulates downstream tasks into language completion tasks with well-designed prompts. Liu et al. (2021a) compose a comprehensive and systematic survey on recent developments of prompt-based techniques, and Saunshi et al. (2021) formulate a solid mathematical framework to explain why prompt-based methods would work for downstream tasks. Based on the survey, we introduce prompt-based learning that forms the basis of our exploration of taking PLMs as automatic re-annotators for promoting the quality of datasets from two aspects: (1) prompt types and (2) their applications related to our task re-annotating the relation extraction datasets.

According to their form, prompts can be divided into two styles: cloze-style and prefix-style. Cloze-style prompts (Cui et al., 2021; Petroni et al., 2019b) require PLMs to fill the intentionally masked span in a semi-completed sentence, and the prediction is made by the filling decisions or ranking made by PLMs. It is typically used to handle classification tasks which have clear restrictions or objectives, such as relation extraction or sentiment analysis. Conversely, prefix-style prompts (Li and Liang, 2021; Lester et al., 2021) encourage PLMs to generate a continuation of the given sentence. It is more suitable for those tasks involving the text generation, such as text summarization or question answering. Methods used to compose prompts can be categorized into manual template-based and automatic generated prompts. The former type of prompts is generated by manually crafted templates based on human understanding of the context of the task. For instance, as humans know that the relation mentions usually appear between the subject and object mentions in the text, inserting the mask token between the mentions of subject and object would be the most rational way to compose cloze-form prompts retrieving the possible relation held in the context. The latter type of prompts is automatically generated, searched or tuned with a small fraction of the downstream data. Considering the flexibility of expressions in natural language, generated prompts can more accurately derive task-related knowledge from PLMs to reinforce downstream performance. However, accompanying this strength, an automated template may make models relatively prone to overfitting compared to a manual crafted template because the strong implications in the prompt could easily mislead PLMs. This drawback is especially worth noting when we apply prompt-based methods for rechecking problematic annotations.

As we investigate the automatic re-annotator in the context of relation annotation, prompt applications in relation extraction could be very relevant. Relation extraction is the task of predicting the underlying relation between the given subjective and objective entities in the context. Chen et al. (2021) identify two major challenges of applying the prompt-based method to the relation extraction task: (1) the difficulties in prompt engineering caused by the enlarged label space of relations, and (2) the elusive importance of the tokens appeared in the context. To overcome these two obstacles, they developed the KnowPrompt framework that constructs the learnable prompt for relation extraction with virtual template words and answers words. They further inject the knowledge of entity and relation via marking entity spans by wrapping the entity mentions with special markers such as `[E]`. Correspondingly, Han et al. (2021) leverage the technique of prompt composition to form the prompt with abundant entity type information. The prompt composition technique is to compose the final prompt by synthesizing several sub-prompts based on logic rules. For instance, to extract the relation between ”Google” and ”Alphabet” with the context ”Google became a subsidiary of Alphabet”, we can compose the complete prompt as ”The [MASK] Google [MASK] the [MASK] Alphabet” from the sub-prompts ”The [MASK] Google”, ”The [MASK] Alphabet”, ” Google [MASK] Alphabet”. The first two sub-prompts enable the PLMs to associate the supplementary knowledge about the entity ”Google” and ”Alphabet” independently before guessing their relation.

2.2 Annotation Noise

Annotation noise in the form of inconsistent or incorrect labels appears to be common in most machine learning datasets. Kusendová (2005) reckon that the noise can follow from unclear or insufficient instructions, unclear contexts, bias on the part of the annotator, or deviations from what the instruction writers expect. Nowadays, training neural network models usually requires vast amounts of annotated data, so this issue has become increasingly prominent in machine learning. The annotators of these datasets often lack expertise in the related domain because the most commonly used method for composing large-scaled supervised data involves crowdsourcing the annotation with the platforms like Amazon Mechanical Turk (Crowston, 2012), and the annotation is done by cheap labour rather than well-paid expert annotators. Specifically, the platforms distribute a small fraction of the whole annotation task with brief instructions to the non-expert crowd workers and then merge the partial annotations together to form the massive dataset. Since the crowd workers may have their own annotation standards, the quality of crowdsourced datasets is worrisome due to potential inconsistencies and errors and endangers the robustness of machine learning systems. Considering these facts, we believe an automatic pipeline for denoising the annotations in datasets is a worthwhile and meaningful way to resolve the bottleneck in machine learning. Furthermore, since PLMs are pre-trained on unlabeled data in an unsupervised learning fashion without any human intervention, applying PLMs to improve the data quality is an appealing method for reducing the uncontrollable human factor in machine learning.

In Section 2.2.1, we introduce the annotation noise in the context of classification tasks from four perspectives: definition, taxonomy, sources and its downstream influence, which specify the context of our investigation of automatic re-annotation. In Section 2.2.2, we describe plausible methods to learn on imperfect datasets without alternating the annotation noise. Compared to noise-tolerant learning, we believe that revising error annotations with PLMs brings more interpretability and insights for handling the annotation noise.

2.2.1 Analysis of Label Noise

Based on the previous work on the label noise (Sharou et al., 2021; Larson et al., 2020; Frénay and Verleysen, 2014; Beck et al., 2020), we briefly demonstrate the label noise involved in classification tasks from the following four facets:

Definition of Annotation Noise For supervised multi-class classification, each sample corresponds to a true class, but the identification of this class would be passed into a noise process to become the observed labels presented to the classification models. Observed annotations may be different from the true labels of samples. In this process, the compromised annotations are called *label noise* (Angluin and Laird, 1987), in contrast to feature noise, which is the perturbation of feature values. However, as noisy labels often have a relatively lower probability of occurrence in their vicinity than the normal labels, the mislabelled examples may be defined as the anomalies or outliers of the distribution of their assigned class in some cases. Therefore, anomaly detection (Schölkopf et al., 1999; Hayton et al., 2000; Schölkopf et al., 2001; Hoffmann, 2007; Chandola et al., 2009) and outlier detection (Barnett, 1978; Sebert, 1997; Zhou et al., 2021b; Hodge and Austin, 2004; Niu et al., 2011; Hawkins, 1980; Winkens et al., 2020) are of great relevance to our task. For instance, Winkens et al. (2020) motivate us to combine the cross-entropy loss with a contrastive loss to enhance the performance of the automatic re-annotator, and we further deliberate the sampling strategy of contrastive learning via a novel distant-based criterion. The framework developed by Zhou et al. (2021b) is also based on contrastive learning and PLMs, detecting the out-of-distribution in relation extraction by the Mahalanobis distance (McLachlan, 1999) of the hidden representations of examples in the penultimate layer. However, our task, re-annotating problematic annotations, is significantly distinguished from the out-of-distribution detection, which intends to solve the problems caused by the different distributions of training and real-world test data.

Taxonomy of Annotation Noise Although the taxonomy of annotation noise is potentially large, we only define two types of annotation noise here to reduce the scope of the investigation: annotation inconsistency and annotation error. Most of the time, the expression of inconsistency and error can be interchangeable, but we would give narrow definitions for them in the context of our project. Annotation inconsistency is the particular annotation that is divergent from the annotations shared by examples with similar context (Larson et al., 2020; Hollenstein et al., 2016; Qian et al., 2021; Li et al., 2020a), but the inconsistent annotation is not necessarily incorrect. For instance, the relation between ”James” and ”Bob” in the context ”James has a son called Bob”, could be annotated as `father` or `family_member_of` rationally. However, if in most similar cases, we choose to annotate the example with the most precise plausible relation, then `family_member_of` label here may be taken as inconsistent. In contrast, annotation error is a broad concept, referring to annotations that are counter to common sense or that contradict the given context Reiss et al. (2020); Suzuki et al. (2017); Matousek and Tihelka (2017); Haverinen et al. (2011); Bryant (2019). Therefore, in the former example, neither label (`father` or `family_member_of`) is an annotation error.

Sources of Annotation Noise Typical sources of annotation noise are: (1) The annotators do not have sufficient information or knowledge to successfully complete the annotation task Hickey (1996); Brodley and Friedl (1999); Pechenizkiy et al. (2006), which is not uncommon when, for example, medical or legal text data is annotated. (2) The inconsistency and errors are introduced in a crowd-sourcing annotation scenario, when a large number of non-experts are involved in the annotation process (Larson et al., 2020; Snow et al., 2008; Raykar et al., 2010; Yuen et al., 2011). (3) The target of the annotation task is ambiguous or subjective, such as annotation for image classification or medical analysis (Grivas et al., 2020; Malossini et al., 2006; Smyth, 1996; Fornaciari et al., 2021). (4) There are distractions during annotation or problems in the design of the annotation interface, such as lack of feedback mechanism (Sculley and Cormack, 2008). Intuitively, automatic re-annotation is expected to mitigate the second and fourth sources of annotation noise effectively. The first source may be solvable by PLMs pre-trained on domain-specific data (Lee et al., 2020; Feng et al., 2020; Li et al., 2020b), but we leave this for future exploration.

Influence on Model Performance The negative impact of annotation noise can be considerable. Experiments conducted by Larson et al. (2020) show that any type of inconsistency in crowd-sourced data would downgrade model performance in slot-filling, though different types of inconsistency may have a different level of impacts. Khayrallah and Koehn (2018) empirically study the impact of diverse annotation noise in a parallel corpus for machine translation and reveal that neural machine translation models are more error-prone to annotation noise than statistical machine translation methods. Chen et al. (2019) suggest that the accuracy on the Test set could be used as the quadratic function to evaluate the noise ratio in the dataset if the annotation noise can be categorized into the symmetric noise (van Rooyen et al., 2015). Alt et al. (2020) and Northcutt et al. (2021a) prove that annotation noise in the Test set significantly misleads the process of model evaluation and selection. Additionally, Northcutt et al. (2021a) argue that less powerful models with fewer parameters or simpler model architecture have more resistance and regularization to robustly learn on the data with asymmetric distribution of noise than large models with more advanced representation learning capability. Hess et al. (2020) present a mathematical explanation for the deterioration of softmax classifiers caused by annotation noise by reformulating a softmax classifier as K-means clustering and deducing the relation between prediction distortion and annotation noise based on Lipschitz Continuity Sohrab (2003).

2.2.2 Noise-tolerant Learning

Instead of improving the annotation quality like automatic re-annotators, noise-tolerant learning intends to minimize the negative impact of the annotation noise during the training period, typically by learning to treat the labels with different degrees of credibility. The most commonly used techniques for mitigating the perturbation of noisy data are soft-labeling (Thiel, 2008) and curriculum learning Soviany et al. (2021); Portelas et al. (2020); Wang et al. (2020); Bengio et al. (2009). Though noise-tolerant learning empirically leads to improved robustness for handling annotation noise, our work improves its fundamental component of suspicious annotation detection with an enhanced PLM-based detector and the interpretability of its black-box learning process with annotation correction.

The core idea of applying soft-label to relieve the noise in training data is to avoid heavily and blindly relying on the observed labels. For example, Thiel (2008) show that the benefit of soft-label in terms of improving noise resiliency of the classifier compared to the hard labels. Liu et al. (2017) construct the soft-label of examples in the noisy distant-supervised relation extraction dataset by jointly considering both the credibility of observed labels and entity-pair correlation in the context, which results in excellent improvement of the model performance. Similarly, Algan and Ulusoy (2021) derive the soft-label from the features of examples in training data and gradually update the soft-label with meta-objective at the beginning of each training iteration, where the meta-objective is obtained by cleaning a small fraction of training data.

The main intuition behind curriculum learning for noise-resistant learning is to let the model learn on the trustworthy data with a large learning rate first and then subtly adjust the parameters based on possibly noisy data. MentorNet proposed by Jiang et al. (2018) is a vivid example to exemplify this idea. The noise-resistant learning procedure involves two paired neural networks, MentorNet and StudentNet. MentorNet learns the curriculum that can help StudentNet focus on the training examples with a relatively high probability of being correctly labelled. The curriculum taught by MentorNet is initially learnt from a tiny dataset with checked labels and then iteratively deliberated based on the learning feedback of StudentNet. Northcutt et al. (2021b) further take MentorNet as the baseline for exploring confident learning in the general domain. Similarly, Zhou et al. (2021a) leverage the curriculum learning based on the data parameters to produce noise-resilient keyword spotting models. To enhance the learning outcomes, Higuchi et al. (2021) utilize the time ensemble of the model and data augmentations to generate pseudo labels for composing noisy data as the harder curriculum for models to learn to handle annotation noise.

2.3 Improving Annotation Quality

Aside from noise-tolerant learning, we could also directly improve or examine the quality of annotation. According to the degree of automation, we categorize the methods for improving annotation quality into three classes: (1) Annotation Process Improvement, that is to manually optimize the factor and steps in the process of annotation, (2) Annotation Inconsistency Detection, that is to spot suspected labels or indicate the reliability of labels based for downstream manual or automatic correction, and (3) Annotation Error Correction, that is to automatically correct the mislabeled examples.

In Chapter 4, we introduce a novel PLM-based approach for Annotation Inconsistency Detection, and in Chapter 5 we further develop contrastive methods for Annotation Error Correction based on cross-validation. Our proposed models have two significant properties that enable them to stand them apart from previous work in Annotation Inconsistency Detection and label rectification: (1) We are the first to leverage the distribution of contextualized embedding from PLMs to indicate the annotation inconsistency and correct the label errors. (2) Our proposed methods do not rely on any explicit linguistic knowledge, pre-defined rules, or cleaned data, which enable them to be generalized to different domains.

2.3.1 Improving the Annotation Process

Improving the process of collecting the annotations is the most straightforward way to alleviate annotation noise, but it usually requires more human labour and increases annotation cost. To reduce the noise caused by non-expert annotators, we could follow previous work in choosing the candidates of annotator who pass a qualification test, systematically training annotators, or cyclical annotation (Roit et al., 2020; Li and Liu, 2015; Alex et al., 2010). To alleviate the noise raised by internal inconsistency between multiple annotators, we could overlap a small fraction of their annotation work to measure their agreement or repeatedly collect the annotations of each example from multiple annotators and then aggregate their decisions comprehensively (Hovy et al., 2013; Passonneau and Carpenter, 2014; Parde and Nielsen, 2017; Nowak and Rüger, 2010; Jamison and Gurevych, 2015). As for the noise introduced by intricate annotation objectives, we may reformulate the annotation targets or equip the annotation platform with an annotation assistant based on active learning (Dobbie et al., 2021; Nghiem et al., 2021; Weeber et al., 2021). The noise due to the distraction during the annotation period also could be improved by monitoring if the annotators could correctly label the probe examples with known golden annotations (Oppenheimer et al., 2009).

2.3.2 Detecting Annotation Inconsistency

There have been extensive investigations in Annotation Inconsistency Detection for a better understanding of the distribution of annotation noise. Early work in this area focussed on detecting inconsistent part-of-speech tags (Abney et al., 1999; Eskin, 2000; Matsumoto and Yamashita, 2000; Nakagawa and Matsumoto, 2002; Matsumoto and Yamashita, 2000; Ma et al., 2001). Since then, the statistical and rule-based methods have been effective in correcting the corpus for other syntactic prediction tasks, such as semantic role labeling (Dickinson and Lee, 2008), and dependency parsing (Dickinson, 2010; Dickinson and Smith, 2011). However, the relation extraction task that we investigated as the semantic task is comparatively complicated to detect the inconsistency in its context because of the flexibility of semantic expression. For instance, detecting inconsistency in multi-word-expression or word sense datasets needs more powerful models. Dligach and Palmer (2011) screen out the annotations that are worth rechecking based on the in word sense data. They regard the examples that are repeatedly predicted as suspicious by both the machine tagger based on support vector machine and the ambiguity detector based on trainable probabilistic classifier as the inconsistent annotations. Hollenstein et al. (2016) propose an algorithm based on the ranking of absolute frequency and entropy of the label distribution to automatically spot inconsistencies in multiword expression and supersense tagging datasets. Qian et al. (2021) develop an automatic inconsistency detector for the task-oriented dialogue dataset, MultiWOZ, based on scripts using regular expressions. Our work is, to the best of our knowledge, the first attempt to detect annotation inconsistencies in the relation extraction domain.

2.3.3 Correcting Annotation Error

Annotation Error Correction is a further step from Annotation Inconsistency Detection, demanding the models to not only identify the suspect labels but also to suggest a rational modification at the same time (Zhang et al., 2015; Nicholson et al., 2015; Bhadra and Hein, 2015). As one of the recent advances in label correction, Wu et al. (2021) develop a meta-learning framework that first approximates the soft label of each example in datasets under the guidance of the small meta dataset with cleaned labels, and then derive the meta learner for annotation correction from the meta-process of soft label estimation. Similarly, Zheng et al. (2021) also carry out the annotation correction process on a small set of data with checked labels as the meta-process and develop the meta-learning based framework composed of two network models, where the meta-model is for correcting the noisy annotations, and the main model is for exploiting the rectified label. The two network models in this framework are collaboratively trained as a bi-level optimization problem. Conversely, Zou et al. (2021) propose an unsupervised ensemble framework for annotation correction without the requirement of checked golden data. They first aggregate the annotations for holistic examples in the dataset by expectation-maximization algorithm, then screen out the hard case to form the targeting dataset with a two-step filtering approach, and finally apply an Adaboost classifier trained on the low-risk remaining dataset to predict the label corrections on the target dataset. In conclusion, aside from being the first to explore the Annotation Error Correction in relation extraction, we also present an appealing research direction of replacing the meta-learning process of label correction with the powerful prior knowledge in PLMs. In the future, we will also explore whether screening out the most suspicious annotations first can aid our proposed automatic annotation rectification.

Chapter 3

Tasks and Data

Annotation Inconsistency Detection and Annotation Error Correction are the two most vital tasks of automatic re-annotation. To clarify the scope of this dissertation, we give both conceptual and mathematical definitions of these two tasks. We also discuss how Pre-trained Language Models can set the foundation for successful re-annotation. With clear objectives, we describe the details of two kinds of datasets we used for conducting the experiments: target datasets for learning and revised datasets for evaluation.

3.1 Re-Annotation in Relation Extraction

Relation Extraction (RE) is the task to predict the semantic relations between given subjective and objective entities, namely head and tail entities, in the context Wang et al. (2021); Aydar et al. (2020); Cui et al. (2017). A typical RE example can be to discern the relation between the head entity ”SpaceX” and the tail entity ”Elon Musk”, given the sentence ”SpaceX was founded in 2002 by Elon Musk”. Ordinarily, RE datasets contain a textual context and spans of head and tail entities in the context and have pre-defined types of plausible relations. Some datasets also provide the auxiliary information of entities, such as Named Entity Recognition (NER) types. Hence, annotation in RE is the process of chosing relations between head and tail entities in the context from a collection of relation types. Relation Extraction is an indispensable component for composing the knowledge graphs Li et al. (2019) that are useful to various downstream NLP applications such as question answering (Dubey, 2021; Saffari et al., 2021; Sen et al., 2021) and dialogue system (Liu et al., 2021b; Chaudhuri et al., 2021; Gao et al., 2021a).

The re-annotation task presupposes that any annotation observed in datasets is not necessarily the true labels of the example in reality because of the annotation noise (Section 2.2.1). Therefore, re-annotation in RE is to re-examine the annotated relations, suggest their credibility, and recommend better relations if possible.

To formally explain the task, we also present the mathematical description of RE re-annotation. Let $\mathcal{R}$ denote the set of relation types, $\mathcal{A}$ denote all annotations, sub and obj denote the subjective and objective entities (head and tail entities) respectively, and c denotes the context of each example. $\mathcal{R}$ contains finite n types of relation t, namely $\mathcal{R} = \{t_i\}^n$. Given context c_i, there is a true valid relation $r_i(sub_i, obj_i) \in \mathcal{R}$ holding between entities sub_i and obj_i. However, due to the possible annotation noise, we can only see $r'_i(sub_i, obj_i) \in \mathcal{R}$ which is the observed relation in datasets. Each example in the datasets with its annotation can be denoted as $a_i(r'_i(sub_i, obj_i), c_i) \in \mathcal{A}$. Thus, re-annotation in RE is to testify whether $r = r'$or reveal the true relation r, based on the observed r', overall annotations $\mathcal{A}$ and implicit real-world knowledge. The corresponding tasks are: Annotation Inconsistency Detection and Annotation Error Correction.

3.1.1 Definition of Annotation Inconsistency Detection

Annotation Inconsistency Detection (AID) is the task to verify if the annotation of each example is consistent with other annotations. AID on Relation Extraction datasets aims to detect whether the observed relation of each example is consistent with other examples with similar entities and similar contexts.

AID could also be conducive for downstream model training in various aspects. It helps a human re-annotator start by looking at the most error-prone annotations to save time. The models may benefit from differentiating the reliable data and unreliable data during training in the curriculum learning fashion (Soviany et al., 2021; Wang et al., 2020), or by adjusting the weights of training data (Wang et al., 2019).

In this project, we define the AID as the binary classification task with the target $y \in [0,1]$ where 1 means the annotation is consistent and 0 means the annotation is inconsistent. For instance, there is a RE example with the context c ”Alan Turing died in 1954” and the observed relation r'(`Alan Turing,1954`) as `date of birth`. AID models are trained to predict $y = 0$ to indicate it as inconsistent, if we have sufficient reasons to believe $r' \neq r$ based on the context c, observed relation r'(`Alan Turing,1954`) and the global distribution of all annotations.

3.1.2 Definition of Annotation Error Correction

Annotation Error Correction (AEC) task is considered to be more complicated than AID task. It not only testifies the annotations but also attempts to predict the true labels for annotations identified as incorrect simultaneously. As for Relation Extraction datasets, AEC models aim to validate the annotated relations between entities in the context and rectify the invalid relations.

Unquestionably, the latest models with millions of parameters could easily capture the representations from the data on an unprecedented level. However, a growing body of research revealed that the data quality is a non-negligible factor that significantly downgrades the model performance or misleads the evaluation (Larson et al., 2020; Khayrallah and Koehn, 2018; Northcutt et al., 2021a). Hence, AEC is expected to become an appealing alternative for manually enhancing data quality. It is common to take human annotators hundreds of hours to revise the large-scale datasets, while AEC systems would vastly shorten the revising time and reduce the cost.

In this dissertation, we define AEC as the multi-class classification task with the target classes that is the same as the set of pre-defined relations $\mathcal{R}$. For example, if an observed relation $r'(\texttt{Alan Turing},\texttt{1954})$ in the context c of "Alan Turing died in 1954" is `date of birth`, AEC models are expected to predict the true relation $r = \texttt{date of death}$, where $r \in \mathcal{R}$.

3.1.3 Basis of Automatic Re-Annotation

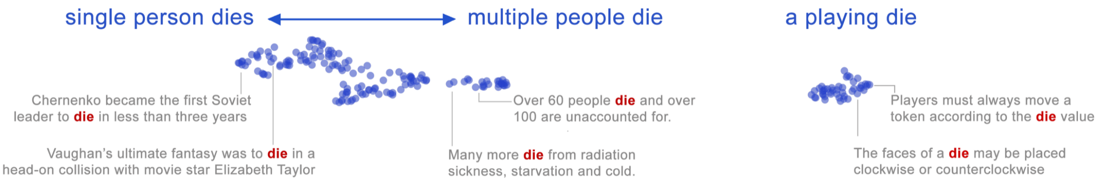


Figure 3.1: Word embeddings from Pre-trained Language Models can capture subtle semantic variations in the context. Hence, Pre-trained Language Models raise novel possibilities in automatic re-annotation. Figure from Reif et al. (2019).

The essential idea of automatic re-annotation is to contrast the query annotation with other annotations in a similar context through models. As shown in Figure 3.1, Pre-trained Language Models (PLMs) demonstrate extraordinary sensitivity to the subtle variations in context. Therefore, we believe it is possible to acquire sufficiently distinguishable relation representations for automatic re-annotation based on the embeddings by PLMs.

Mathematically, Saunshi et al. (2021) uncover the key factors for successfully applying PLMs to the task of interests by reformulating the PLM-based classification tasks into sentence completion tasks. They give a criterion, Definition 3.2 in their paper, for judging whether a downstream classification task $\mathcal{T}$ can be considered as a natural task, namely an analogous sentence completion task, with regards to PLM-based embeddings $\Phi \in \mathbb{R}^{d \times V}$, where V is the vocabulary size. If we assume $p \in \Delta \mathcal{C}$ to indicate probability distribution over context $\mathcal{C}$, and $p_{\cdot|c}$ to denote the true conditional distribution over words in vocabulary $\mathcal{W}$ in given context c, the criterion can be formulated as the inequality:

$$\min_{\mathbf{v} \in \text{row-span}(\Phi), \|\mathbf{v}\|_\infty \leq B} \mathrm{l}_{\mathcal{T}}(\{\mathbf{p}_{\cdot|c}\}, \mathbf{v}) \leq \tau \tag{3.1}$$

, where $\mathrm{l}_{\mathcal{T}}$ is the 1-Lipschitz surrogate (Mairal, 2013) to the classification loss of task $\mathcal{T}$, $\{\mathbf{p}_{\cdot|c}\}$ denotes a language model, and τ and B are two constrains. Through the detailed proofs, they give an intuitive interpretation of τ and B: (1) τ indicates the ambiguity of downstream task measured by Bayes error (Franklin, 2005), and (2) B is inversely proportional to the probability mass of the set of indicative words. Thus, the less ambiguous downstream task which mainly involves frequent words will have smaller τ and B. Qualitatively, Saunshi et al. (2021) suggest two positive factors that help PLM to solve downstream tasks: (1) the PLM-based embeddings can capture the needed semantic meaning in the context and (2) tasks of interest is solvable by distinguishing words with obviously different meanings.

Based on the enlightenment, we develop the frameworks for AID and AEC tasks described in Chapter 4 and Chapter 5 respectively.

3.2 Datasets in Relation Extraction

To comprehensively study the automatic re-annotators, we need two types of datasets: (1) target datasets with unchecked annotations and (2) their corresponding revised datasets with cleaned annotations.

Target datasets is the real-world data with observed annotations that are not necessarily correct or consistent. Therefore, the first usage of target datasets is to ask the re-annotators to learn on them, and then assess if re-annotators can detect inconsistencies or correct errors automatically. In this case, we combine all data splits with unchecked annotation, namely original Train, Dev and Test sets together as the new Train set for our proposed re-annotator. The second usage of target datasets is for downstream evaluation, namely to testify whether the state-of-the-art (SOTA) RE models can benefit from the data denoised by our proposed annotation corrector. In this case, we follow the original data splits to conduct the downstream RE experiments. More details about the downstream evaluation will be introduced in Section 5.3.

On the other hand, we assume the revised versions of target datasets as the golden standard of revision. If we consider the rechecked annotations in revised datasets as ground truths, we can evaluate our proposed re-annotators by comparing their predictions with the manual revisions in revised datasets with classification metrics. Therefore, we derive the Dev and Test sets with checked annotations for our proposed re-annotator from the revised datasets.

Since we decide to investigate the re-annotation in RE, we choose the popular sentence-level RE dataset TACRED (Zhang et al., 2017a) and document-level RE datasets DocRED (Yao et al., 2019b), as two target datasets. The sentence-level dataset TACRED already has two revised versions called TACRev (Alt et al., 2020) and Re-TACRED (Stoica et al., 2021). Nevertheless, to the best of our knowledge, there is no existing manual revision of document-level datasets DocRED. Therefore, to study the new challenges posed by the context length and inter-sentence complexity, we manually re-annotated a subset of DocRED ourselves, called Re-DocRED.

3.2.1 Target Datasets

TACRED

TACRED [1] (Zhang et al., 2017a), The TAC Relation Extraction Dataset, is one of the largest and most widely used datasets for sentence-level RE task, containing examples from web text to news articles from TAC Knowledge Base Population (TACKBP) challenges. Examples in TACRED cover 41 positive relation types which is identical to TACKBP challenges (e.g., `per:title_of`) and one negative type (`no_relation`) if no defined relation is held between given head and tail entities. These examples are created by combining available human annotations from the TACKBP challenges and crowd-sourcing. However, as shown by Alt et al. (2020) and Stoica et al. (2021), TACRED is a typical dataset with pervasive annotation errors and inconsistencies. Since TACRED originally have 68,124 Train samples, 22,631 Dev samples and 15,509

[1] https://nlp.stanford.edu/projects/tacred/

Test samples, there are in total 103, 738 examples for training the sentence-level re-annotators.

DocRED

DocRED [2] (Yao et al., 2019b), Document-Level Relation Extraction Dataset, is a document-level RE dataset derived from Wikipedia and Wikidata. DocRED requires reading multiple sentences in a document to extract entities and infer their mutual relations by common-sense reasoning and aggregating contextual information of the document. Compared to the TACRED dataset, the examples in DocRED datasets generally exhibit more complex inter-sentence relations. They can be more intricate than the existing relation extraction (RE) methods that focus on extracting intra-sentence relations for single entity pairs. Moreover, DocRED has 96 valid relation types which is also siginificantly more than TACRED.

DocRED contains 132,375 entities and 56,354 relational facts on 5,053 human-annotated Wikipedia documents. We are the first to examine the annotation quality and spot the annotation deficiency on DocRED. As shown in Table 3.1, we eliminate some examples with multiple relations to reduce the ambiguity while training, and there are 50,503 examples in the Train set for document-level re-annotators.

Dataset	Split	#Example	#Positive	#Negative	#Relation
TACRED	Train	103,738	19,247	84,491	42
TACRev	Dev	1,263	596	667	40
	Test	1,263	628	635	39
Re-TACRED	Dev	5,364	3,596	1,768	36
	Test	5,365	3,608	1,757	36
DocRED	Train	50,503	50,503	0	96
Re-DocRED	Dev	206	206	0	55
	Test	205	205	0	55

Table 3.1: The statistics of target datasets TACRED and DocRED and their revised datasets TACRev, Re-TACRED, and Re-DocRED datasets. The example is negative if no defined relation is held between head and tail entities.

[2] https://github.com/thunlp/DocRED

3.2.2 Existing Revised Datasets

TACRev

TACRev (Alt et al., 2020) was the first investigation on the annotation noise in the TACRED dataset. Linguists re-examined the most challenging 5,000 examples in TACRED, and 2,526 of them were revised from the original labels, of which roughly 57% of the negative labels were modified into the positive labels. They proved that the performance ceiling of previous SOTA models on TACRED datasets is largely due to the annotation noise, showing that 4 SOTA models can improve 8% absolute F_1 test score by evaluating on refined TACRev dataset.

The TACRev dataset only releases the 2,526 revised examples without the annotations proved to be correct. We first shuffle the 2,526 revised examples and then evenly split them into Dev set and Test set (Table 3.1). The Dev set contains 40 relations including `no_relation`, while Test set contains 39 relations.

Re-TACRED

Compared to TACRev, Re-TACRED (Stoica et al., 2021) is the crowd-sourced version of TACRED revision instead of being relabeled by several linguists. The entire TACRED dataset was rechecked by an improved and cost-efficient crowd-souring annotation strategy with a quality control mechanism. During the process of relation definition refinements intended to resolve the ambiguous relation definition in TACKBP, the authors further introduced new relations (e.g. `org:-member` as the inverse relation of `org:members` and `org:subsidiaries`) and rename several initial relation (e.g. `per:alternate_name` to `per:identity`). Since the automatic re-annotators are impossible to predict the new relations by merely learning on the vanilla TACRED dataset without these new relations, we delete all examples with the newly introduced relations in Re-TACRED to accommodate our task. Trained and evaluated on the Re-TACRED dataset, an average of 14.3% improvement of F_1-score could be observed, which indicates that Re-TACRED essentially enhanced the annotation quality and could assess relation extraction models more faithfully.

Similarly, we exploit the Re-TACRED to appraise to what extent our proposed automatic re-annotator can relabel the dataset with higher consistency by merely learning on noisy data. We recompose the Re-TACRED dataset for upstream evaluation following the same strategies on TACRev. The dataset includes 10,729 examples that have been relabeled and still held relations already existed in the original TACRED and

TACKBP relation set. To make the results on Re-TACRED comparable to TACRev, we only keep the revised examples, shuffled and then evenly split into Dev set with 5264 examples and Test set with 5365 examples that covered 36 different relations.

3.2.3 New Revised Dataset: Re-DocRED

We create the first dataset containing manual re-annotations in document-level RE, called Re-DocRED. It is a revised subset of DocRED, containing the revision of examples with challenging annotation inconsistencies and errors.

Since the labels of the Test set of DocRED are non-public, the Re-DocRED is derived from the Train and Dev sets of DocRED, which only have 50503 relation facts. Re-DocRED includes 411 re-annotations totally and is evenly split into Dev and Test set with the coverage of 55 relation types (Table 3.1). Re-DocRED dataset is sampled and annotated in two phases:

Data Selection We firstly follow the similar data selection strategy proposed in TACRev (Alt et al., 2020) to make maximum use of our restricted human labour. The potentially challenging examples for annotation are screened out according to the disagreement of the predictions by multiple state-of-the-art RE models. Specifically, we fine-tuned the CorefBERT (Ye et al., 2020) with BERT-base, ATLOP (Xu et al., 2021) with RoBERTa-large, and SSAN (Xu et al., 2021) with RoBERTa-base and RoBERTa-large on the Train set and contrast their predictions with the human annotations on the Dev set of DocRED to figure out the error-prone labels from Dev examples. Unlike the criteria applied in TACRev, which is to select the misclassified examples by at least half of the models, we empirically observe that if human annotations are different from all these four model predictions, the annotations are likely to be incorrect or inconsistent on DocRED. This automatic data selection procedure narrows the scope for human revision to 560 examples.

Manual Annotation The selected examples are firstly validated by a recent graduated undergraduate student in Linguistics and then inspected by the author of this dissertation, a master by research student in Linguistics with a bachelor degree in Computer Science. The validation and inspection are independently conducted via the online annotation platform based on INCEpTION [3] (Klie et al., 2018). It is an open-source

[3] https://inception-project.github.io

semantic annotation platform built by UKP Lab at TU Darmstadt. The manual validation and inspection procedures roughly take 35 and 10 hours, respectively. Consequently, 411 high-quality re-validated examples from DocRED Dev set are selected to compose the Re-DocRED dataset, including 57.5% examples with revised labels and 42.5% examples with original labels.

Chapter 4

Annotation Inconsistency Detection

Annotation Inconsistency Detection testifies whether each annotation is consistent with other majority annotations in a similar context. We explore various relation representation methods and neighbouring consistency measurements for the zero-shot inconsistency detectors in relation extraction. The results suggest that our proposed credibility score combined with the relation prompt effectively detects inconsistencies.

4.1 Overview

This chapter will discuss the Annotation Inconsistency Detection (AID) based on the static relation representations directly from the Pre-trained Language Models (PLMs) without fine-tuning. The detailed definition of the AID task and the mathematical notations are introduced in Section 3.1.1.

To achieve the best results from zero-shot AID models, we first explore the prompt-based and entity-based methods to acquire the relation representations with full utilization of prior knowledge in PLMs. Based on the semantic sensitive PLM-based relation representations, we approximate the neighbouring consistency with the K-Nearest Neighbours algorithm and a novel credibility score jointly computed by the distance and trustworthiness of retrieved neighbours. Consequently, we propose two approaches for detecting annotation inconsistencies: (1) compare the observed annotation with the majority annotations of K-Nearest Neighbours, and (2) consider the observed annotation with credibility lower than a certain threshold as inconsistent.

The empirical experiments show that both prompt-based and entity-based methods can result in the PLM-based relation embedding with needed semantic sensitivity for distinguishing inconsistent relation labels. Our proposed credibility scores combined with proper relation representations demonstrate impressive capability in detecting annotation inconsistencies, reaching the binary F_1 up to 92.4 on TACRev, 92.1 on Re-TACRED and 72.5 on Re-DocRED. Through the visualization, we reveal that overconfident prompts could downgrade the sensitivity and specificity of AID models.

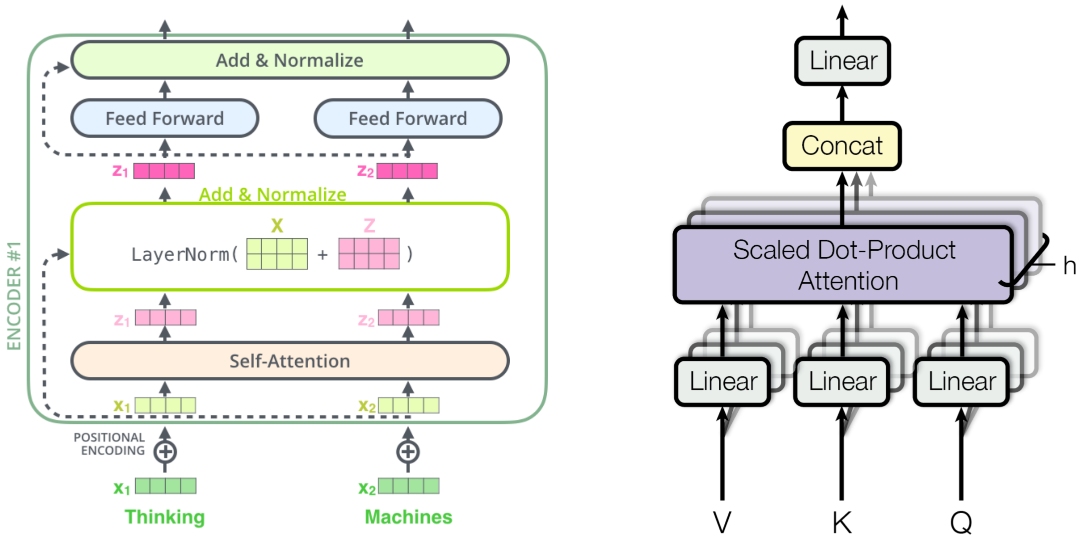
4.2 Methodology

- Section 4.2.1 briefly introduces the technical details of PLMs, such as types of special tokens and pre-training tasks, for better understanding our proposed relation representation methods.
- Section 4.2.2 describes prompt-based and entity-based methods for obtaining PLM-based relation representation.
- Given noise-sensitive relation representations, Section 4.2.3 proposes the K-Nearest Neighbours and credibility score based AID models.

4.2.1 Pre-trained Language Models

Large-scale Pre-trained Language Models (PLMs) as the contextualized word embedding techniques have become the dominant workhorse in NLP because in various downstream tasks, they lead to a better performance than traditional fixed word embedding methods such as one-hot embedding Glove (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013). The strength of PLMs could be roughly attributed to: (1) massive learnable parameters allow models to capture richer interactions between each token in the context and (2) large-scale pre-training endows models with abundant common and linguistic knowledge. Recent studies (Petroni et al., 2019a; Gao et al., 2021b; Cao et al., 2021; Yao et al., 2019a; Hewitt and Manning, 2019a; Shin et al., 2020; Davison et al., 2019; Jiang et al., 2020b; Talmor et al., 2020) imply that properly designed prompts for formatting the text input could explicitly leverage intrinsic knowledge from PLMs to enhance few-shot learning tasks.

Since only BERT was used in this project, this section will only introduce the details of architectures and pre-training tasks proposed by Devlin et al. (2019). The architecture of its backbone model is identical to the multi-layer bidirectional Transformer encoder described by Vaswani et al. (2017). The encoder typically is composed of a stack of self-attention layer and position-wise fully connected feed-forward layer with residual connections (Figure 4.1a). The self-attention layer exploits the multi-head mechanism with scaled dot-product attention (Figure 4.1b) to capture the mutual interaction within the input sequence.



(a) The architecture of the multi-layer bidirectional Transformer encoder.

(b) The architecture of multi-head attention, the basic component of self-attention layer.

Figure 4.1: The architecture of the backbone model of BERT. Figures from Vaswani et al. (2017).

Several special tokens are introduced to the vocabulary to support BERT in diverse downstream tasks: **[CLS] token:** special classification token used to acquire the sentence-level representation for sequence classification tasks (e.g. sentiment classification); **[SEP] token:** Separator token used to differentiate the consecutive sentences; **[MASK] token:** mask token used to replaced the tokens indented to be masked during pre-training.

The pre-training data of BERT is two large-scale document-level corpora, BooksCorpus (Zhu et al., 2015) with 800M words and English Wikipedia with 2,500M words. As shown in Figure 4.2, BERT is pre-trained with two unsupervised tasks:

- **Masked Language Model (MLM):** Some percentage of the input tokens are masked randomly, and then the model learns to predict the masked tokens similarly to the Cloze-filling task. As the [MASK] token usually does not appear in the downstream tasks, only 80% of the time masked tokens are replaced by [MASK] token, while 10% of the time by random tokens and 10% of the time

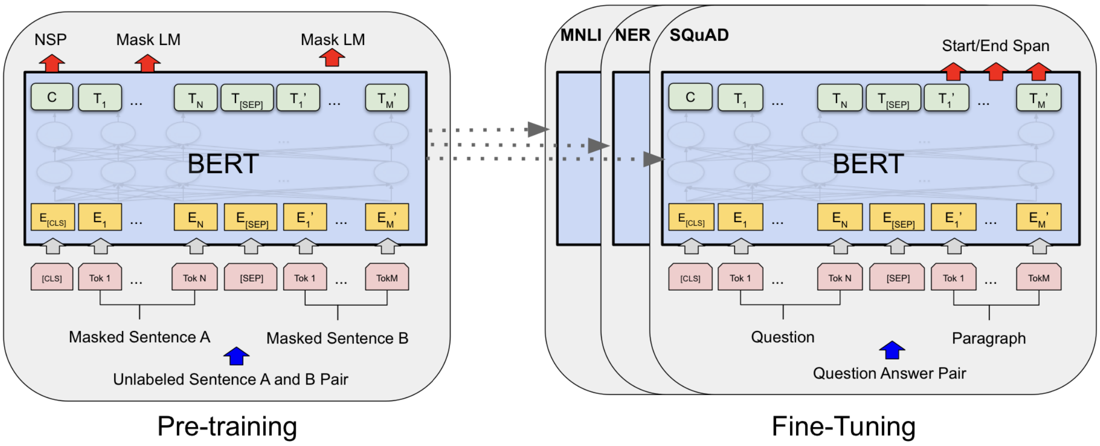


Figure 4.2: The PLMs framework includes two stages: (1) pre-training on general purpose tasks such as Masked Language Model on large-scale corpus; (2) fine-tuning for the downstream tasks on domain-specific datasets. Figure from Devlin et al. (2019).

by unchanged tokens, to avoid mechanical memorization.

- **Next Sentence Prediction (NSP):** Given sentence `A`, the model learns to predict its next sentence `B`. 50% of the time `B` is the actual next sentence that follows `A`, while 50% of the time `B` is not. Argued by Liu et al. (2019b), this task is not as informative as the MLM task.

4.2.2 Relation Representation Methods

The basis of AID in Relation Extraction (RE) is to acquire the relation representations with desired semantic meaning from the context and entity mentions. As mentioned in Section 3.1.3, the main assumption of the AID task is that the representations of inconsistent samples should have different observed relations with their neighbours in the embedding space. Thus, we explore two methods to derive the relation representation from PLMs without fine-tuning: (1) prompt-based method, and (2) entity-based method. In this zero-shot scenario, guaranteeing their capability of fetching needed prior knowledge from PLMs is challenging and consequently determines the final performance of AID models. However, as PLMs perform only the inference stage, it will be viable if the computational resource is limited.

Prompt-based Representations

As described in Section 2.1.2, PLMs acquire abundant prior knowledge from a large corpus through pre-training tasks MLM and NSP. However, the gap between pre-training tasks and downstream tasks restricts knowledge transferability. Hence, Prompt-based Learning with PLMs becomes the new paradigm in natural language processing, narrowing this gap by formulating downstream tasks into mimic LM tasks and introducing inductive words. (Liu et al., 2021a). The conventional approach trains PLMs to directly predict the output y given the input context c by $P(y \mid c)$, and y is from a different label set without explicit connection with the vocabulary of PLMs. Nevertheless, prompt-based methods modify the input c using a template into a new textual string prompt c' with unfilled slots, and then the predictions are acquired by observing the discrete guesses or processing latent representations of PLMs on these masked slots. In the AID task, we explore methods to acquire the relation representation from the latent representations of masked slots. If the prompt c' is properly designed, this approach naturally narrows the gap between the pre-training and downstream domain-specific tasks, such as AID. It enables our models to readily perform few-shot, or even zero-shot learning with minimum loss of prior knowledge of PLMs learnt from the large-scale pre-training data.

The prompt we designed for AID comprises two parts, the context given in each example and a template following it. The context part is a single sentence on TACRED-based datasets or a document on DocRED-based datasets. It relieves the disparity between the pre-training MLM task and our target AID task by providing explicit context. The prompts contain the mentions of head and tail entities and a $[MASK]$ token in the position m for deducing the relation representation $\mathbf{e}_r$ by:

$$\mathbf{e}_r = \mathbf{h}(\text{[MASK]}) = PLM(\text{prompt})_m$$

, where $\mathbf{h}(\text{[MASK]})$ is the last hidden states in the masked position obtained from a PLM.

As shown in Table 4.1, we explored five different variations from three types of template:

- **Fixed Templates**: The first and second methods fix a single template for all relation types.

- **Hand-written Templates**: The third method manually assigns every relation type with hand-written templates.

- **Generated Templates**: The fourth and fifth methods generate the templates for each example. They ask the PLMs to fill one token between head mention and [MASK], and another one token between [MASK] and tail entity mentions, with the following steps: (1) the example likes r'(`Bill Gates,Microsoft`) is converted into the template `Bill Gates [MASK] Microsoft`; (2) If the PLMs fill the masked slot with the token `,`(comma), we fill the [MASK] in the previous template with `,`(comma) and insert a new [MASK] after this decoded token to convert the previous templates into `Bill Gates, [MASK] Microsoft`; (3) We repeat above steps until three consecutive tokens are filled between head and tail mentions, like `Bill Gates, CEO of Microsoft`; (4) We then mask the middle token of the generated three consecutive tokens to become the prompt used for acquiring relation representations, like `Bill Gates, [MASK] of Microsoft`. The fourth method is to directly fill each masked slot with the top candidate given by PLMs, while the fifth method randomly selects one of the top three candidates to fill each slot.

Templates	Examples
[HEAD] is [MASK] of [TAIL]	Bill Gates is [MASK] of Microsoft
[HEAD] [MASK] [TAIL]	Bill Gates [MASK] Microsoft
[Human-tuned template for each relation type]	Bill Gates works as [MASK] of Microsoft
[Auto template filled with top candidate]	Bill Gates, [MASK] of Microsoft
[Auto template filled with top 3 candidates randomly]	Bill Gates was [MASK] of Microsoft

Table 4.1: The examples of different templates for prompt-based relation representation. In the examples, subjective entity is ”Bill Gates” and objective entity is ”Microsoft”.

Entity-based Representations

Aside from deriving the relation representations from masked tokens as prompt-based methods, fully exploiting the entity-wise embeddings is an alternative to acquiring the relation representations. Generally, salient notation of head and tail entity mentions could provide considerable hints for PLMs to discern the authentic relation between them. We follow the entity representation techniques discussed by Zhou and Chen (2021), to obtain the relation embedding $\mathbf{e}_r$ by concatenating the contextualized embeddings $\mathbf{e}_{sub}$ and $\mathbf{e}_{obj}$ of subject and object entity as follows:

$$\mathbf{e}_{entity} = \mathbf{h}_e = PLM([x_0, ..., x_n])_e$$
$$\mathbf{e}_r = [\mathbf{e}_{sub} : \mathbf{e}_{obj}]$$

, where x_i is the input token, $\mathbf{h}_\mathrm{e}$ is the last hidden states of the input token in the position of *entity pointer e*. Considering the entities usually appear more than once in the context with the markers in different places $[\mathbf{e}_0, ..., \mathbf{e}_n]$ on DocRED-based datasets, we use the average of all the hidden states of entity markers $\mathbf{h}_{e_n}$ as the entity representation:

$$\mathbf{e}_{entity} = avg([\mathbf{h}_{e_0}, .., \mathbf{h}_{e_n}])$$

As exemplified in Table 4.2, the investigated techniques for entity representations include:

- **Entity position**: It takes the first tokens of head and tail entities as the *entity pointers* without modifying the given context.
- **Entity marker** (Zhang et al., 2019; Soares et al., 2019): It introduces two special tokens pair `[H]`, `[/H]` and `[T]`, `[/T]` to enclose head and tail entities, which indicates the positions and spans of entities. The tokens `[H]` and `[T]` are *entity pointers*.
- **Entity marker (punct)** (Zhou et al., 2020): It is similar to the entity marker but replaces the special tokens pair by punctuation existing in the vocabulary, such as # and @. The punctuation in front of each entity mention is the *entity pointer*.
- **Entity mask**(Zhang et al., 2017b): It adds the new special tokens `[SUBJ-TYPE]` and `[OBJ-TYPE]` to mask the spans of the head and tail entities, where `TYPE` is substituted by their named entity types. The special tokens `[SUBJ-TYPE]` and `[OBJ-TYPE]` are *entity pointers*.
- **Typed entity marker** (Zhong and Chen, 2020): It is similar to the entity marker but further provides the information regarding entity types.
- **Typed entity marker (punct)**: It is similar to the Entity marker (punct) but further provides the information of named entity types.

Input Formats	Examples
Entity position	Bill Gates founded Microsoft.
Entity marker	[H] Bill Gates [/H] founded [T] Microsoft [/T].
Entity marker (punct)	@ Bill Gates @ founded # Microsoft #.
Entity mask	[SUBJ-PERSON] founded [OBJ-CITY].
Typed entity marker	<S:PERSON>Bill Gates </S:PERSON>founded <O:CITY ....
Typed entity marker (punct)	@ [person] Bill Gates @ founded # ! city ! Microsoft #.

Table 4.2: Different input formats to highlight the entity mentions for entity-based relation representations.

4.2.3 Neighbouring Consistency

Neighbouring consistency is a vital clue for detecting abnormal annotations. Intuitively, the annotation reliability of each example may be judged by the similarity between its embedding and the embeddings of other examples with the same label. Hence, we investigate the neighbouring consistency based on the distribution of all relation embeddings encoded by PLMs in the representation space without fine-tuning PLMs on downstream tasks. We propose two approaches in order to capture annotation inconsistencies based on neighbouring correspondence: (1) K-Nearest Neighbours (KNN) and (2) Kernel Density Estimation (KDE) based detectors.

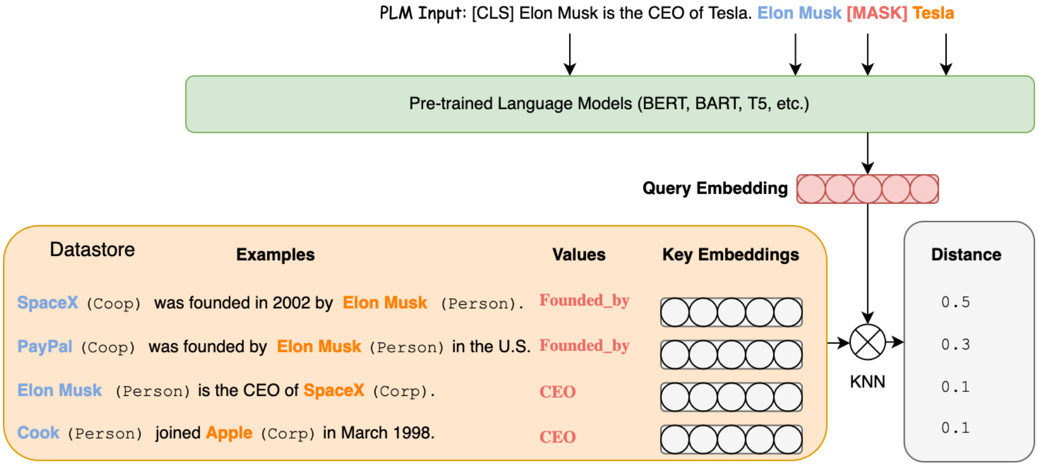

PLM Input: [CLS] Elon Musk is the CEO of Tesla. Elon Musk [MASK] Tesla
Pre-trained Language Models (BERT, BART, T5, etc.)
Query Embedding
Datastore
Examples
Values
Key Embeddings
SpaceX (Coop) was founded in 2002 by Elon Musk (Person).
Founded_by
PayPal (Coop) was founded by Elon Musk (Person) in the U.S.
Founded_by
Elon Musk (Person) is the CEO of SpaceX (Corp).
CEO
Cook (Person) joined Apple (Corp) in March 1998.
CEO
KNN
Distance
0.5
0.3
0.1
0.1

Figure 4.3: An example of retrieving neighbours of a query with prompt-based representation based on the K-Nearest Neighbours algorithm.

The K-Nearest Neighbours (KNN) algorithm (Fix and Hodges, 1989) is a non-parametric method for classification by a plurality vote of its neighbours. As suggested by Grivas et al. (2020), analyzing the nearest neighbour annotations retrieved by KNN may effectively reveal potential annotation inconsistencies. We exploit the framework similar to (Khandelwal et al., 2020) to retrieve neighbours of the query embedding by squared Euclidean distance (Figure 4.3). The query embedding is the relation representation of the example in the Test and Dev set, acquired by PLMs with prompts or entity markers. The datastore contains the keys k and values v : (1) keys k are the relation representations of all instances in the Train set, and (2) values v are observed relations. Therefore, given a query, the KNN algorithm would return a list of nearest neighbours sorted by the distance metrics that measure the similarity between the query and stored keys. Then, we further propose the following vote-based and credibility-based methods to verify the query annotations by analyzing the retrieved neighbours.

Vote-based Detection

The vote-based detection follows the conventional approach of KNN-based classification that determines if the annotation of each example is consistent with the majority voting of their neighbours. That means, if the annotation of the example is the same as the most frequent annotations of its neighbours, the vote-based models predict the annotation as valid.

Credibility-based Detection

Since the vote-based methods only rely on the closest neighbours and consider them as isolated, we propose a credibility-based approach for testifying the annotation consistency. Inspired by the inference function in Khandelwal et al. (2020), we define a novel credibility score combining both the local neighbouring and the global distributional information.

Assuming each instance in retrieved neighbours $\mathcal{N}$ has relation r_{n_i} and relation representation $\mathbf{e}_{n_i}$, the credibility score ψ_i of each annotation with relation r_i and relation embedding $\mathbf{e}_i$ could be computed by first aggregating probability mass s_i across all its neighbours with the same annotation and then rescaling to $\psi \in [0,1]$:

$$s_i = \sum_{(r_{n_i},\mathbf{e}_{n_i})\in\mathcal{N}}^{r_i=r_{n_i}} f_K(\mathbf{e}_{n_i})exp(-d(\mathbf{e}_{n_i},\mathbf{e}_i)), \;\; s_i \in \mathcal{S} \tag{4.1}$$

$$\psi_i = norm(s_i) = \frac{s_i - min(\mathcal{S})}{max(\mathcal{S}) - min(\mathcal{S})} \tag{4.2}$$

, where exp is the exponential function.

The d in Equation 4.1 is the normalized squared Euclidean distance function, computed as follows:.

$$d(\mathbf{e}_{n_i}, \mathbf{e}_{r_i}) = \frac{|\mathbf{e}_{n_i}, \mathbf{e}_{r_i}|}{max(\mathcal{D})}, \quad |\mathbf{e}_{n_i}, \mathbf{e}_{r_i}| \in \mathcal{D} \tag{4.3}$$

It gathers the local information by measuring the distance between an example and its neighbours with the same relation type. We normalize the distance to $d(\mathbf{e}_{n_i}, \mathbf{e}_{r_i}) \in [0,1]$ to prevent $exp(-d(\mathbf{e}_{n_i}, \mathbf{e}_i))$ from being too close to 0, which helps keeping the information from remote neighbours. The closer neighbours result in higher $exp(-d(\mathbf{e}_{n_i}, \mathbf{e}_{r_i}))$, which means the closer neighbours contribute more significantly to the credibility score.

The f_K in Equation 4.1 is the Probability Density Function using a Gaussian Kernel (Murphy, 2012). We utilize Kernel Density Estimation (KDE), a method to estimate the probability density function, to capture the global information from the overall embedding distribution of each relation type. If the probability density $f_K(\mathbf{e})$ is high, the embedding is near the centroid of distribution of all examples with identical relation types. Intuitively, we believe that the neighbours with higher probability density regarding their annotated relation type are more trustworthy for contributing to the credibility score. Noted the $\mathcal{K}$ as the standard normal distribution function, the KDE function f_K could compute the probability density of an example with the embedding e and relation r as the relation type t as follows:

$$f_K(\mathbf{e}) = f_K^t(\mathbf{e}) = \frac{1}{|\mathcal{E}_t|h} \sum_{\mathbf{e}_i \in \mathcal{E}_t} \mathcal{K}(\frac{\mathbf{e} - \mathbf{e}_i}{h}) \tag{4.4}$$

, where h is the bandwidth, and $\mathcal{E}_t$ is the set of embeddings of all examples with relation type t.

Finally, the prediction y_i is obtained by comparing the credibility score ψ_i with a threshold $\beta \in [0,1]$.

$$y_i = \begin{cases} inconsistent, & \psi_i < \beta \\ consistent, & \psi_i \geq \beta \end{cases} \tag{4.5}$$

4.3 Evaluation

The performance of the proposed AID approaches is evaluated based on the consistency between human revisions and model predictions. Hence, by regrading the decisions made by human revisers as the ground-truth, the performance of different relation embedders and the results of AID models can be quantified with rank-based metrics and classification metrics, respectively.

Rank-based Metrics Rank-based Metrics are able to measure the sorted retrieval results obtained by KNN models directly. The relation representation methods are expected to map the embeddings of examples with the same true annotations to be close together. Hence, given a particular relation representation as input, the examples with the true annotation of the query example in its neighbours are expected to be ranked as high as possible by the KNN models. The Hit@1, Hit@5, Hit@10, and Mean Reciprocal Rank (MRR) (Radev et al., 2002) are used to evaluate the relation representation methods.

Classification Metrics Classification Metrics are sensible to evaluate the model performance on AID tasks because the AID is a binary classification task. Accuracy and binary F_1 score (Grishman and Sundheim, 1996) are utilized to assess if the AID models are in agreement with human revisers. Accuracy intuitively reflects the proportion of predictions identical to the manual revisions. At the same time, F_1 score is the harmonic mean of the precision and recall, demonstrating a better measure of the incorrectly classified cases.

4.4 Experiments

- Section 4.4.1 presents the implementation details of all experiments in AID.
- Section 4.4.2 demonstrates the experimental results regarding different relation representation techniques described in 4.2.2.
- Section 4.4.3 shows the outcomes of different inconsistency detecting strategies mentioned Section 4.2.3.

4.4.1 Implementation Details

Pre-trained Language Model The off-the-shelf backbone model we used is the pre-trained BERT-BASE-CASED[1] from Hugging Face. It has the maximum length for the input sequence as 512 tokens, and the tokenizer is based on WordPieceWu et al. (2016).

[1] https://huggingface.co/bert-base-cased

The multi-head encoder has 12 attention heads, and the dropout probability between adjacent hidden layers is 0.1. It totally contains 12 hidden layers, and both the embedding and hidden layers have dimensions of 768. Therefore, the dimension of the relation representations acquired by prompt-based methods (Section 4.2.2) would be 768, whereas 1,536 of the embeddings obtained by entity-based methods.

K-Nearest Neighbours Searching The K-Nearest Neighbours retrieval is implemented with Faiss[2] (Johnson et al., 2017). Faiss is the library including several methods for high-performance similarity search in the embedding space. As for the Faiss implementation, each instance is assumed to be a vector embedding and indexed by an integer, where the vectors can be compared with squared Euclidean distances. Examples that are similar to a query are those that have the vector embeddings with the lowest squared Euclidean distance with the query vector.

Handling Long Sequences As the contexts of the examples in DocRED is the entire document, the input sequences sometimes would exceed the maximum input length of BERT-BASE-CASED model. Therefore, we incrementally truncate the long sequences with the following modes in order until the input lengths meet the requirement:

- **Mode 1**: It remains the sentences between the first and the last sentences that contain the mentions of arbitrary entities.
- **Mode 2**: It remains the sentences including the mentions of arbitrary entities.
- **Mode 3**: It iteratively decreases the number of sentences containing entities until the input length is shorter than the maximum input length.

Datasets Both the vote-based KNN detector and the credibility detector require the information of the existing relation embedding distribution. All 103,738 examples of TACRED and 50,503 examples of DocRED are used to produce the embedding datastore. As for KNN-based methods, the vector representations of relations are used as the keys, and the original annotations of examples are regarded as the values for the neighbour search. As for KDE-based methods, the relation embeddings contribute to the training of the KDE model of their belonging classes according to the original annotations in TACRED and DocRED. We evaluate the performance of different AID

[2] `https://github.com/facebookresearch/faiss`

systems based on the manual revisions provided in TACRev, Re-TACRED and Re-DocRED: If the manual revision is identical to the original annotations, the target label is `True` and vice-versa. First, we use the Dev and Test sets of TACRev, both containing 1,263 human revisions for systematical evaluation on relation encoders. Then, additionally to TACRev, we take 5,364 and 5,365 revisions from the Dev and Test sets of Re-TACRED, and 206 and 205 revisions from the Dev and Test sets of Re-DocRED as the ground-truth to optimize the hyperparameters of inconsistency detectors and comprehensively evaluate their performance. More details of the datasets used to conduct the experiments are presented in Section 3.2.

Hyper-parameters The hyper-parameters of KNN-based methods include the number k of voting neighbours, and we only explore the cases of $k = 1$ and $k = 3$. The hyper-parameters of credibility-based methods are the number of retrieved neighbours for computing the credibility score, the bandwidth h of the KDE model in Equation 4.4 and the threshold β of the credibility-based classifier in Equation 4.5. The number of retrieved neighbours was manually set to be 250. By tuning hyper-parameter with Bayesian optimization (Snoek et al., 2012) on the Dev sets of TACRev, Re-TACRED and Re-DocRED, the bandwidth h was optimized as 0.25, and the threshold β was set as 0.5 for all datasets.

Baselines To properly evaluate the performance of different relation representation methods, we regard the three sentence-level relation representations as to the baseline models: (1) the sentence representation by feeding the last hidden states of the `[CLS]` token into the pre-trained BERT pooling layer for the NSP task; (2) the sentence embedding by average pooling over the hidden states of all tokens in the sequence; (3) the aggregated embedding by feeding the last hidden states of all tokens into the maximum pooling layer.

Experimental Environments The following experiments were conducted on a single GeForce GTX 1080 Ti with 12GB graphic memory and CUDA version of 11.0. The proposed models are implmented with the PyTorch 1.9.0[3] (Paszke et al., 2019) , Transformers 4.3.3[4] (Wolf et al., 2020), Scikit-learn 1.0.1[5] (Pedregosa et al., 2011) ,

[3] `https://pytorch.org`

[4] `https://github.com/huggingface/transformers`

[5] `https://scikit-learn.org/`

Numpy 1.20.3[6] (Harris et al., 2020) and GPU-based Faiss 1.7.1.

Representation Method		Hit@1	Hit@5	Hit@10	MRR
Baseline Sentence-level	Pooler Layer	41.1	64.4	70.3	51.5
	Average Pooling	41.3	67.4	77.1	53.0
	Max Pooling	40.0	67.6	77.6	52.8
Entity-based	Entity position	57.2	86.7	92.7	70.0
	Entity marker	54.2	88.0	94.1	68.7
	Entity marker (punct)	56.6	**89.5**	**94.5**	70.3
	Entity mask	57.9	88.9	93.7	70.7
	Typed entity marker	55.2	88.9	94.2	69.7
	Typed entity marker (punct)	56.8	87.5	93.2	70.2
Prompt-based	[HEAD] is [MASK] of [TAIL]	65.1	88.5	93.6	75.8
	[HEAD] [MASK] [TAIL]	**67.2**	88.4	93.5	**76.9**
	[Human-tuned template for each relation type]	5.3	8.1	8.4	6.4
	[Auto template filled with top candidate]	2.3	4.5	6.8	3.7
	[Auto template filled with top 3 candidates randomly]	10.2	23.8	31.1	16.7

Table 4.3: The accuracy and binary F_1 of the relation representations obtained by sentence-level, prompt-based, entity-based embedders on TACRev Test set.

4.4.2 Representation Methods of Relation

Table 4.3 shows the Hit@1, Hit@5, Hit@10 and MRR of the KNN query results based on the embedding acquired by sentence-level embedders, entity-based embedders and prompt-based embedders. The sentence-level representation approaches are regarded as the baselines. Generally, the average pooling of all token embeddings in the context is the best representation method among the sentence-level encoders, reaching the Hit@1 of 41.3 and MRR of 53.0.

As the embeddings of the newly introduced marker tokens are randomly initialized without downstream fine-tuning, it is noticeable that using the punctuation to mark the entity spans slightly outperforms the methods with new special tokens. For instance, the ENTITY MARKER (PUNCT) that marks the entity with punctuation likes # increases the Hit@1 by 2.4 and MRR by 1.6 compared to the ENTITY MARKER that marks the entity with special tokens like `[H]`. Similarly, TYPED ENTITY MARKER (PUNCT) enables the KNN-based detector to achieve higher Hit@1 and MRR than TYPED ENTITY MARKER. In contrast, the ENTITY POSITION and ENTITY MASK without including any special tokens or new language patterns that PLMs have not been exposed to in

[6] `https://numpy.org`

the pre-training tasks are the most competitive methods in the entity-based relation encoders. The KNN-based detector with ENTITY MASK technique reaches the highest Hit@1 of 57.9 and MRR of 70.7 among all entity-based embedders, outperforming the best baseline by 40.2 % in Hit@1 and 33.4 % in MRR. The results indicate that methods that smooth the learning curve of PLMs lead to better relation representations, which potentially supports the arguments by Saunshi et al. (2021) as briefly described in Section 3.1.3.

In contrast, the appropriate prompts result in even better compatibility between AID tasks and pre-training tasks than the entity-based representation methods. Both [HEAD] IS [MASK] OF [TAIL] and [HEAD] [MASK] [TAIL] templates show the better overall performances than the sentence-level and entity-based encoders. The prompt generated by [HEAD] [MASK] [TAIL] template leads to the optimal result of all relation representations with Hit@1 of 67.2 and MRR of 76.9, exceeding the best entity-based model by 8.8% and the best baseline by 45.1% in term of MRR. However, the human-tuned and automatically searched prompts dramatically downgrade the performance of relation embedders. According to the empirical studies, we find that this degradation could be accused to that the obvious hints in the prompts or contexts make the PLMs to be less sensitive to the annotation inconsistencies. A detailed discussion of the trade-off of relation representations will be presented in Section 4.5.1.

4.4.3 Classification by Neighbouring Agreements

According to the experimental results of different relation representations, we finalize three representation methods for all following experiments: (1) **Relation Prompt**: the prompt generates by the [HEAD] [MASK] [TAIL] template, (2) **Entity marker**: the entity spans are wrapped by `[H]` or `[T]`, and (3) **Entity marker (punct)**: the entity spans are wrapped by `@` or `#`. [HEAD] [MASK] [TAIL] is selected because of its overwhelming performance among prompt-based methods. As the entity-based methods, including the external NER knowledge, do not bring prominent advantages over other methods, we decline those methods to focus on the information that can be directly grabbed by the PLMs from the context. ENTITY MARKER and ENTITY MARKER (PUNCT) are selected because their overall performances are archetypal among entity-based methods, and comparing the influences of different types of markers is of interest.

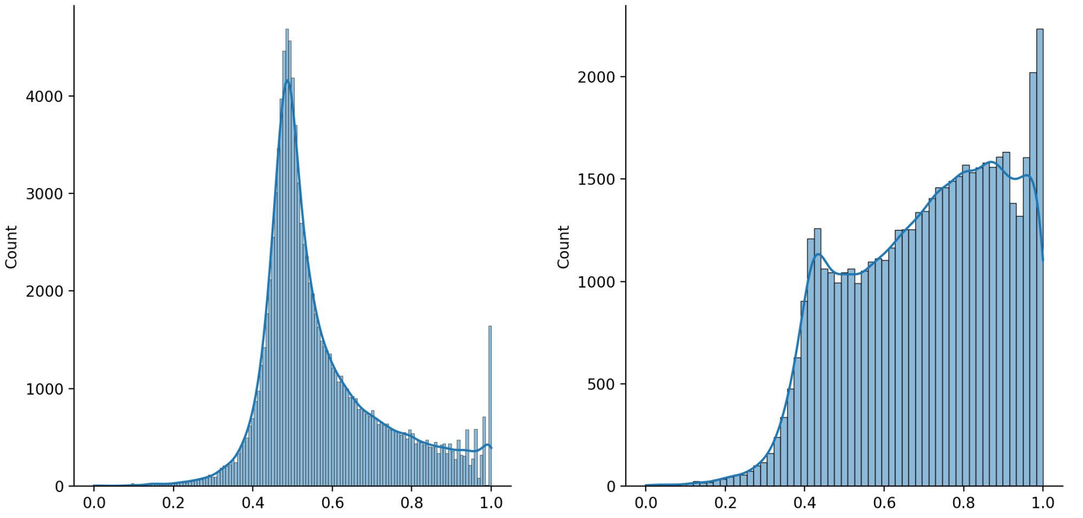
Table 4.4 illustrates the performances of the KNN-based detectors and credibility-

		TACRev		Re-TACRED		Re-DocRED	
Detection Methods	Relation Representation	Acc	F_1	Acc	F_1	Acc	F_1
KNN-based (k=1)	Relation Prompt	31.0	47.4	47.3	64.2	43.4	49.6
	Entity marker	28.4	44.3	45.1	62.2	48.3	55.1
	Entity marker (punct)	26.6	42.0	45.5	62.5	47.8	54.5
KNN-based (k=3)	Relation Prompt	45.8	62.9	64.7	78.5	44.9	56.0
	Entity marker	46.2	63.2	64.2	78.2	49.8	61.7
	Entity marker (punct)	43.0	60.1	63.4	77.6	51.2	63.2
Credibility-based (β = 0.5)	Relation Prompt	**85.9**	**92.4**	**85.3**	**92.1**	59.0	69.1
	Entity marker	78.1	87.7	71.9	83.6	60.0	72.1
	Entity marker (punct)	84.4	91.5	72.6	84.1	**60.1**	**72.5**

Table 4.4: The accuracy and binary F_1 of KNN-based and credibility-based inconsistency detection methods with different relation encoders are evaluated on TACRev, Re-TACRED, Re-DocRED Test set. Credibility-based AID models obviously outperform other methods.

based detectors on the Test set of TACRev, Re-TACRED, Re-DocRED. The KNN-based models with $k = 3$ consistently surpass those with $k = 1$, indicating that the annotation of the closest neighbour may be deceptive. The credibility-based indicators evidently outshine all KNN-based detectors with the increment of 39.7 in accuracy and 29.2 in F_1 score on TACRev, 20.6 in accuracy and 13.6 in F_1 score on Re-TACRED, and 8.9 in accuracy and 9.3 in F_1 score on Re-DocRED at least. It proves that our proposed credibility-based score is more effective in detecting annotation inconsistencies by jointly considering the local geometry of neighbours and the global embedding distributions of each class.

In Figure 4.4, it is clear that the examples with credibility scores under certain threshold form the long tail of the distribution. This phenomenon is adhered to the intuition due to the fact that the inconsistent annotations should never be in the majority of any sound corpus. The adaptive threshold for credibility scores will be left for future explorations.



(a) The distribution of the credibility scores on TACRED.

(b) The distribution of the credibility scores on DocRED.

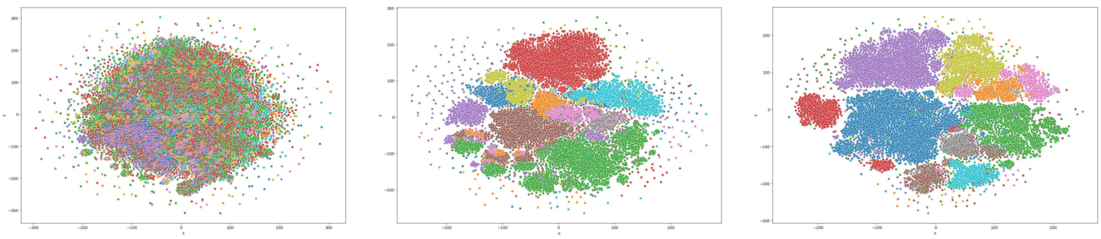
Figure 4.4: The long tails of the distributions of the credibility scores on the entire TACRED and DocRED datasets are apparent when the score is under certain threshold.

4.5 Discussions

4.5.1 Trade-off of Relation Representations

Mindset is a set of assumptions, methods, or notions held by people which are incentives to continually adapt the prior behaviours or choices (Argyris, 2004; Taylor and Gollwitzer, 1995). As observed, the bias of mindset could happen to the PLMs when they encounter prompts with strong implications. It is especially fatal for the AID task because it relies on the neutral prior knowledge of the PLMs to reexamine the annotations with due impartiality. However, the strong indications in the prompts possibly misdirect the detectors to make arbitrary decisions on the ambiguous annotations.

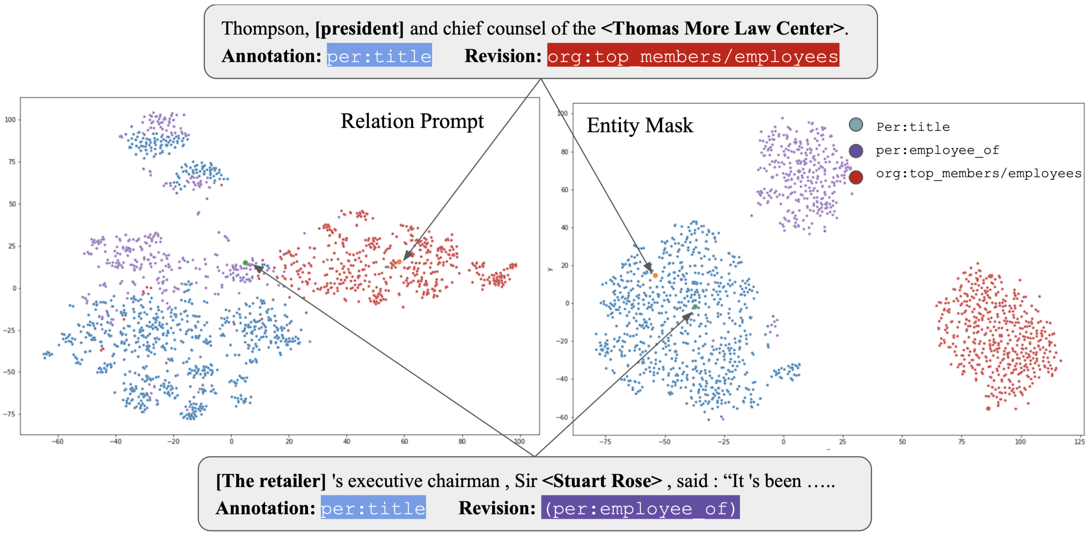
We leverage the T-SNE (van der Maaten and Hinton, 2008) to visualize the relation embedding acquired by different methods we mentioned in Section 4.2.2. As shown in Figure 4.5, the cluster of the embedding belonging to different relation types become more differential with the growth of the prior knowledge in prompt designs. Since the sentence-level methods do not provide any additional knowledge about neither the relation extraction task nor the AID task, its final relation representations almost have not shown any distinguishable cluster. In the sharp contrast, the relation representations obtained by [HEAD] [MASK] [TAIL] and human-tuned template



(a) (b) (c)

Figure 4.5: The T-SNE visualization of the relation representations by sentence-level and prompt-based embeddings, and the colors stands for the labels of examples. **(a):** The relation representations by sentence-level max pooling. **(b):** The relation representations by the prompts with [HEAD] [MASK] [TAIL] template. **(c):** The relation representations by prompts with human-tuned template.

form differential clusters according to relation types. Compared with the clusters by [HEAD] [MASK] [TAIL] templates with a slight overlap between adjoining classes, the clusters by human-tuned templates are separated by clearer boundaries. Counter-intuitively, the AID task actually requires the embedding distribution to meet a subtle balance between distinctive and farraginous. The strong inclination of clustering usually means the detectors would be overconfident with the observed annotations.



Thompson, **[president]** and chief counsel of the **<Thomas More Law Center>**.
Annotation: `per:title` **Revision:** `org:top_members/employees`

[The retailer] 's executive chairman , Sir **<Stuart Rose>** , said : "It 's been …..
Annotation: `per:title` **Revision:** `(per:employee_of)`

Figure 4.6: The T-SNE visualization of two examples with the same incorrect annotation of `per:title` that should be revised into `org:top_members/employees` and `org:employee_of` respectively.

The Figure 4.6 illustrates a typical detection error caused by the strong implication in prompts. There are more obvious three clusters, `per:title`, `org:top_members/employees` and `org:employee_of` clusters, of the relation representations by ENTITY MASK than RELATION PROMPT. Nevertheless, two examples with the same error annotation `per:title` are still located in the inappropriate `per:title` cluster by ENTITY MASK, but they are mapped into the correct `org:top_members/employees` and `org:employee_of` clusters by RELATION PROMPT.

Prompt tuning, as one of the emerging paradigm in NLP, doubtless unleashes the potential of massive pre-trained models (Gao et al., 2021b; Li and Liang, 2021; Zhong et al., 2021). However, our experiments remind that the over artefact prompts may easily lead to the dilemma in the era of feature engineering: the trade-off between generalizability and overfitting.

Chapter 5

Annotation Error Correction

Annotation Error Correction detects suspicious annotations and recommends true annotation for them simultaneously. Instead of zero-shot learning, we fine-tune the error corrector by cross-validation. We introduce the uncertainty to the observed hard label, effectively mitigating the negative impact of annotation noise during cross-validation. We also develop the rank-aware neighbouring encoder and distant-peer contrastive loss to enhance the neighbour awareness of AEC models. Empirically, distant-peer contrastive loss with uncertain soft labels is the optimal configuration of AEC models throughout our study.

5.1 Overview

This chapter will demonstrate the Annotation Error Correction (AEC) based on the dynamic relation representations by fine-tuning the PLM-based neural relation classifier with cross-validation. The comprehensive definition of the AEC task and the formal notations are described in Section 3.1.2.

Compared to zero-shot learning, PLM-based neural relation classifiers fine-tuned by cross-validation enable AEC models to suggest better annotations for suspected examples precisely. However, as AEC models cross-validate on the noisy observed data, to alleviate the noise problems of softmax classifier, we convert the overconfident observed labels in training folds into the labels with uncertainty by Kernel Density Estimation (KDE) and K-Nearest Neighbours (KNN). We also enhance AEC models with neighbouring information based on the rank-aware Transformer encoder and a novel distant-peer contrastive loss.

Empirically, we observe that fine-tuning leads to better AEC performance than zero-shot methods introduced in Chapter 4. Uncertain labels show better properties over the hard labels for fine-tuning AEC models on noisy observed datasets. The neighbouring awareness is also conducive to the AEC models generally. Consequently, the AEC model augmented with distant-peer contrastive loss and Kernel Density Estimation based uncertain labels outperforms other configurations, achieving macro F_1 of 66.2 on TACRev, 47.7 on Re-TACRED and 57.8 on Re-DocRED.

Practically, cleaning the training set of relation extraction datasets with our proposed AEC framework leads to up to 3.6% downstream improvement for state-of-the-art relation extraction models. According to the statistics of needed time per examples, our proposed AID and AEC models beat the human revisers with significantly shortened re-annotation time.

5.2 Methodology

- Section 5.2.1 introduces the cross-validation for fine-tuning the AEC models on observed data.
- Section 5.2.2 proposes the uncertain labeling for relieving the noise problem encountered during cross-validation.
- Section 5.2.3 suggests two methods for injecting the neighbouring information to the learning process of AEC models.

5.2.1 Cross Validation

Though well-designed relation embedders and neighbour-based detectors are proved to be effective in revealing the potential annotation inconsistencies, the static relation representations may not be sufficient to correct the invalid annotations. The credibility-based scores can indicate the consistency of a given annotation, but it is impossible to suggest another annotation. Though the KNN-based detectors are able to speculate the true labels based on the majority voting by retrieved neighbours, they are usually over-sensitive to the noisy neighbours. Especially for the AEC task, as there is no guarantee on the quality of training data, KNN-based classifiers can hardly predict the true label, if the example is surrounding by compromised neighbours.

In order to address the shortcoming of KNN for AEC, we attempt the AEC task by predicting the verified labels for the annotated example with the conditional probability:

$$r = \arg max_{r \in \mathcal{R}} Pr(r | r'(sub, obj), c, \mathcal{A}) \tag{5.1}$$

, which means the AEC models suggest the relation that is the most compatible with both the observed annotation and the internal congruity over the entire dataset.

Cross-validation (Stone, 1977; Tibshirani, 1996; Allen, 1974) uses different portions of the data to test and train a model on different iterations, typically for estimating the model performance in practice or optimizing the hyper-parameters. In contrast, we leverage cross-validation to learn the conditional probability in Equation 5.1 on target datasets with unchecked annotations.

As for the Leave-one-out cross-validation, the entire dataset is usually evenly split into several folds. Then, in each iteration, one fold of data is sequentially selected to simulate the unseen data for hyper-parameter searching and testing, and other folds are merged together to train the models. Inspired by the cross-validation, we explore similar methods to train the PLM-based AEC models.

The neural relation classifier is built by the hidden layer and softmax classifier. The relation embedding vector $\mathbf{e}_r \in \mathbb{R}^d$ acquired by the prompt-based methods (Section 4.2.2) or the entity-based methods (Section 4.2.2), is first fed into the hidden layer with the *ReLU* non-linear activation:

$$\mathbf{h} = ReLU(\mathbf{W}_{proj}\mathbf{e}_r) \tag{5.2}$$

, where $\mathbf{W}_{proj} \in \mathbb{R}^{d \times d}$ is the linear projection matrix and $\mathbf{h}$ represents the hidden states from the hidden layer. Then, based on the hidden representation $\mathbf{h}$, the softmax classifier predicts the conditional probability of relation r given context c and observed annotation r':

$$Pr(r \mid c) = \frac{exp(\mathbf{W}_r\mathbf{h} + b_r)}{\sum_{r' \in \mathcal{R}} exp(\mathbf{W}'_{r'}\mathbf{h} + b'_{r'})} \tag{5.3}$$

, where $\mathbf{W}_r, \mathbf{W'}_{r'} \in \mathbb{R}^{d \times R}$ and $b_r, b'_{r'} \in \mathbb{R}^R$. After training with cross-validation algorithm, the neural classifiers are expected to learn the relation with the maximum probability as the predicted rectified annotation.

Under the AEC setting, we train the models with the targets of unchecked annotations on training folds but expect the models to predict the labels on the held-out fold as the same as the human revised label (Figure 5.1). Statistically, the annotations on training fold $\mathcal{A}_{\text{Train folds}}$ could estimate the characteristics of the overall annotations $\mathcal{A}$ in Equation 5.1 by sampling. Since the training process totally relies on the original annotations in the datasets without being exposed to any kind of revisions, there is no data leakage in the cross-validation.

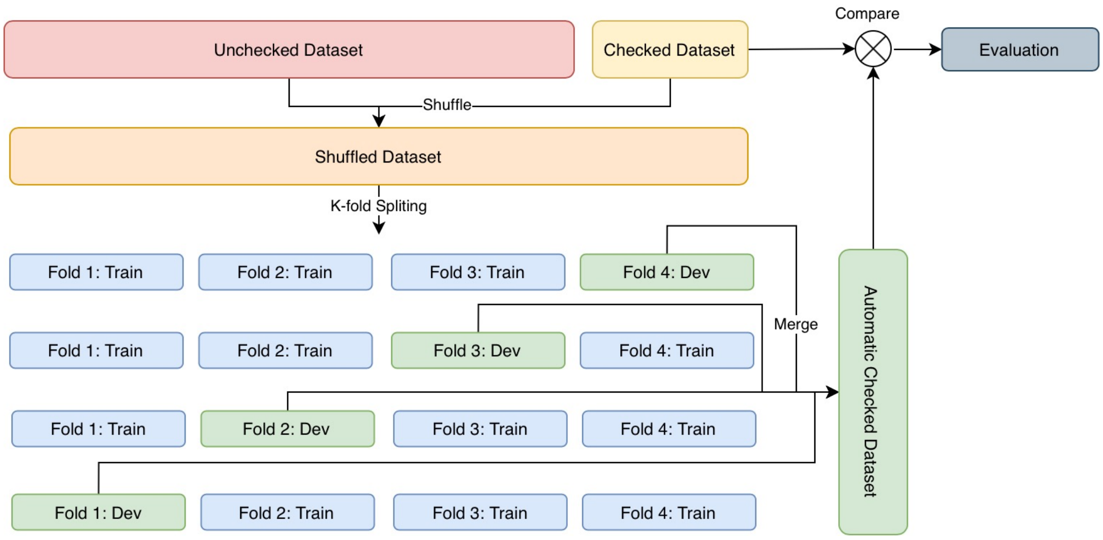


Figure 5.1: The Pre-trained Language Models are fine-tuned by cross validation on the unchecked dataset. The automatically corrected annotations are obtained by merging the held-out Dev sets from all iterations of the cross validation.

As for the AID task, the PLMs already show impressive ability to detect abnormal annotations solely relying on their extensive inner common-sense knowledge learnt by pre-training on the massive general domain corpus. However, the cross-validation could further reinforce this strength to accommodate prior knowledge in PLMs to AEC task in three ways: (1) Fine-tuning endows the PLMs with the both task- and domain-specific knowledge in Relation Extraction; (2) Fine-tuning allows `[MASK]` tokens in prompts and new special tokens in entity-based methods (e.g. `[H]` in ENTITY MARKER) to be adapted for relation verification.; (3) neural relation classifiers give better predictions based on the supervised learning, instead of only relying on the error-prone neighbouring information.

5.2.2 Uncertain Labeling

Although the softmax classifier empirically leads to excellent classification performance, it is also not absolutely robust to noise. Thus, we further introduce uncertainty to labels to avoid cross-validation based AEC model from over trusting the observed labels during training on the target datasets. We first present the mathematical explanation of noise sensitivity of softmax classifier by Hess et al. (2020), and then describe two possible solutions to reduce the inducement of noise based on their theoretical insights.

According to Hess et al. (2020), noise sensitivity of softmax classifier can be mathematically illustrated with Lipschitz Continuity and the relation between softmax classifier and k-means Clustering. First, the Equation 5.2 and Equation 5.3 of neural classifier could be expressed by a more general form:

$$F(x) = \sigma(f_p(x)^\intercal W) \tag{5.4}$$

, where σ stands for softmax function and $W \in \mathbb{R}^{d \times c}$ is the matrix of weights. The f_p denotes the penultimate layer of neural network that maps the n-dimensional input space to c-dimensional probability vector, and c should be the same as the number of valid classes. This formulation omits the expression of bias vector and affine function.

Lipschitz Continuity (Tsuzuku et al., 2018) can theoretically measure the robustness of models by demonstrating the effect of the perturbations of the input. If the function $f : \mathbb{R}^n \to \mathbb{R}^c$ is Lipschitz continuous with modulus L, for every $x_1, x_2 \in \mathbb{R}^n$ it should satisfy:

$$||f(x_1) - f(x_2)|| \leq L||x_1 - x_2||$$

The Lipschitz modulus of function F in Equation 5.4 is decided by $L_p||W||$, where L_p is the modulus of function f_p because the modulus of softmax function is less than one:

$$||F(x_1) - F(x_2)||^2 \leq ||f_p(x_1)^\intercal W - f_p(x_2)^\intercal W|| \leq L_p||W||\,||x_1 - x_2||$$

As for the neural networks, the small Lipschitz modulus, which implies that the adjacent data points have close function values, could also indicate the model robustness by restricting the effect on the classification of small distortions of data points with inequalities.

Hess et al. (2020) prove two theorems revealing the connections between softmax classifier and k-means Clustering:

Theorem 1: *Let the dimension of the penultimate layer d be larger than or equal to the number of classes: $d \geq c - 1$. Assumed a network output is $y = \arg max_k f_p(x)^\intercal W_{\cdot k}$, there exist c class centroids $Z_{\cdot k} \in \mathbb{R}$ with equal distance to the origin, such that every x is classified to the class whose center is nearest in the transformed space:*

$$y = \arg min_k ||f_p(x) - Z_{\cdot k}||^2$$

Theorem 2: *Let Z be the center matrix in Theorem 1 and $x \in \mathbb{R}^n$ be a data point with predicted class k. Assumed that f_p is Lipschitz continuous with modulus L_p, any* *distortion $\Delta \in \mathbb{R}^n$ which changes the prediction of point $\tilde{x} = x + \Delta$ to another class $l \neq k$ has a minimum value of:*

$$\|\Delta\| \geq \frac{\|Z_{\cdot l} - Z_{\cdot k}\| - \|f_p(\tilde{x}) - Z_{\cdot l}\| - \|f_p(x) - Z_{\cdot k}\|}{L_p}$$

These two theorems exemplify that the noise sensitivity may be relieved from three aspects: (1) reduce the Lipschitz modulus of neural networks, (2) maximize the mutual distance of the centroids of different classes, and (3) map x or $\tilde{x}$ close enough to their class centroids.

Considering the annotation errors are widely spread in the datasets and the noise sensitivity of the softmax classifier, we follow the second and third insights to prevent AEC models from unsuspectingly depending on the observed labels in the training folds. Following the concept of reappraising existing labels of an example based on the annotation tendency of its neighbours (Section 4.2.3), we propose two approaches that intentionally introduce the uncertainty to the observed labels in datasets to enable trained models to amend the annotations on the held-out folds prudently (Figure 5.2): (1) let some observed labels adopt the majority labels of their neighbours to map the marginalised x close to proximal class centroids, which may not be their original class centroids, by alternating their class membership, and (2) soft-label with Kernel Density Estimation for maximising mutual distance of the centroids of different classes by assigning probabilistic class membership.

Label Replacement by K-Nearest Neighbours

As discussed in Section 4.2.3, retrieved neighbours presumably hint at the underlying true label of the query example. Therefore, letting a suitable fraction of the labels adopt the true labels predicted by the KNN classifier may introduce the right amount of helpful uncertainty to the Train folds, because some marginalised x after alternating class membership can be considered less noisy by softmax classifiers.

Given the observed relation label r' and neighbouring relations $\{r_{n_0}, r_{n_1} ...\}$, the uncertain label $\bar{r}$ are controlled by the probabilistic switch with a manually appointed threshold $\phi \in [0,1]$ as follow:

$$\bar{r} = \begin{cases} KNN(\{r_{n_0}, r_{n_1} ...\}), & p_u < \phi \\ r', & p_u \geq \phi \end{cases} \tag{5.5}$$

, where KNN is the K-Nearest Neighbours classifier, and p_u is a random value following the uniform distribution in range $[0,1]$. Hence, the threshold ϕ decides the ratio of

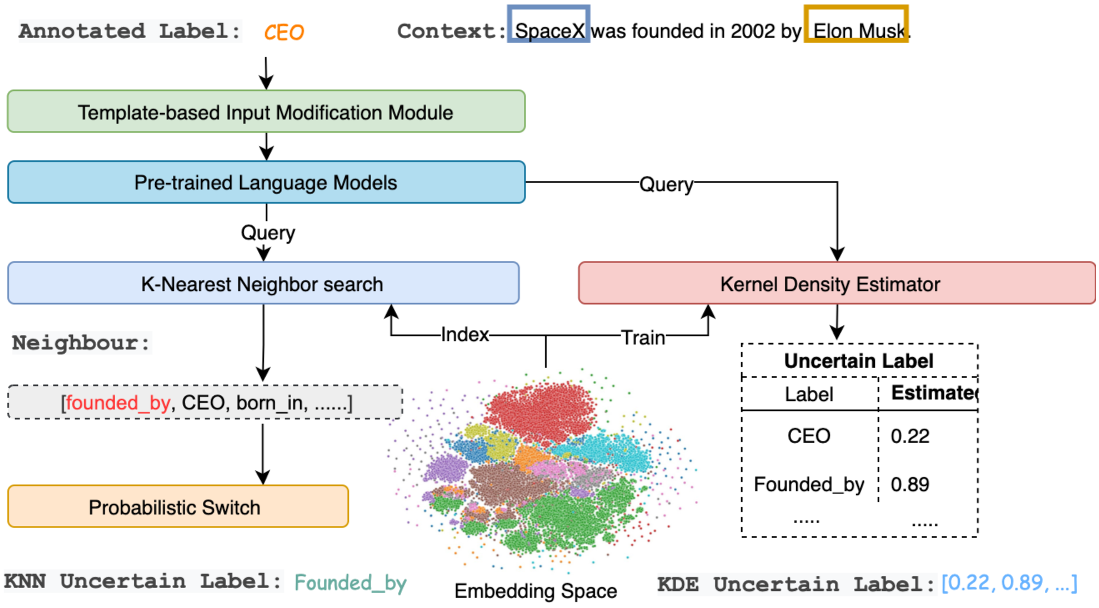

Annotated Label: CEO
Context: SpaceX was founded in 2002 by Elon Musk.
Template-based Input Modification Module
Pre-trained Language Models
Query
Query
K-Nearest Neighbor search
Kernel Density Estimator
Neighbour:
Index
Train
[founded_by, CEO, born_in, ...]
Uncertain Label
Label
Estimated
CEO
0.22
Founded_by
0.89
.....
.....
Probabilistic Switch
KNN Uncertain Label: Founded_by
Embedding Space
KDE Uncertain Label: [0.22, 0.89, ...]

Figure 5.2: The pipeline that converts the certain labels in the datasets into the labels with uncertainty by K-Nearest Neighbours and Kernel Density Estimation.

label replacement in the whole datasets.

Soft-label with Kernel Density Estimation

Soft labels are widely accepted as the paradigm for handling noisy data in supervised learning (Thiel, 2008; Nguyen et al., 2014; Liu et al., 2017; Zhao et al., 2014; Algan and Ulusoy, 2021). In contrast to hard labels where class membership is certain, soft labels express uncertainty in which single label should be assigned. (Galstyan and Cohen, 2007). Combined with the probabilistic cross-entropy loss, soft-label enables the model to consider the supervised signals from multiple possible classes.

Intuitively, we could use the similar idea of soft-label to prevent the AEC models from blindly trusting the annotation in the Train folds. According to Section 4.2.3, given the embedded example and the relation type $t \in \mathcal{R}$, the probability density estimated by Equation 4.4 can be regarded as the likelihood that the example indeed belongs to this relation type t. Therefore, the KDE model with appropriate bandwidth h could convert the one-hot hard label r' on the unchecked training folds into a soft label vector $\tilde{\mathbf{r}} \in \mathbb{R}^{|\mathcal{R}|}$ with its estimated probability density regarding each relation class:

$$\tilde{\mathbf{r}} = \{f_{\mathcal{K}}^{t_0}(\mathbf{e}_r), ..., f_{\mathcal{K}}^{t_n}(\mathbf{e}_r)\}, t_i \in \mathcal{R} \tag{5.6}$$

, where $\mathbf{e}_r$ is the relation embedding, $f^t_{\mathcal{K}}$ is the KDE models with the Gaussian kernel regarding relation type t, and $\mathcal{R}$ is the set of relation types.

5.2.3 Neighbour-aware Correction

The neural classifiers are built by the fully-connected feed-forward layers followed by a softmax layer. It is able to better predict annotation inconsistencies by supervised learning on the entire training sets, in contrast to the KNN classifiers which only rely on the local geometry of the distribution of annotations. The empirical experiments in AID tasks indicate that static relation representations derived with properly crafted prompts or markers from the off-shelf PLMs are semantically sensitive to annotation noise. Fine-tuning the PLMs and neural classifier would further enhance this preponderance by subtly adapting the static relation representation to the relation extraction task. Hence, the neural classifier stacked on top of the PLMs trained with cross-entropy loss by the cross-validation is a strong baseline for the AEC task.

However, a vanilla neural relation classifier is not immediately aware of neighbouring relations when making its decisions. Such information would be helpful since the neighbouring consistency hints the essence of observed annotations as discussed in Chapter 4. We therefore propose to augment the vanilla neural relation classifier into neighbour-aware classifiers with two alternatives: (1) rank-aware neighbouring encoders or (2) contrastive learning with distant-peer positive examples.

Rank-aware Neighbouring Encoder

The KNN classifier treats each retrieved neighbour as an atomic individual but ignores their crucial mutual interactions. Therefore, we regard the query example and retrieved neighbours analogously as the sequential textual inputs and acquire their contextualized embedding with the rank-aware Transformer encoder (Vaswani et al., 2017) (Figure 5.3). The inputs to the Transformer encoder is the sequence of relation representations of both the query example and its neighbours in rank order. The self-attention multi-head encoder acquires the neighbour-aware relation embedding of the query example by jointly aggregating the information from its neighbours and capturing the mutual interaction among neighbours. Meanwhile, the positional encoding utilizes the hints behind the order of its neighbours.

The rank-aware neighbouring encoder intends to replace the vanilla embedding vector $\mathbf{e}_r$ in Equation 5.2 with the neighbour-aware embedding vector $\mathbf{e}'_r$. Each rela-

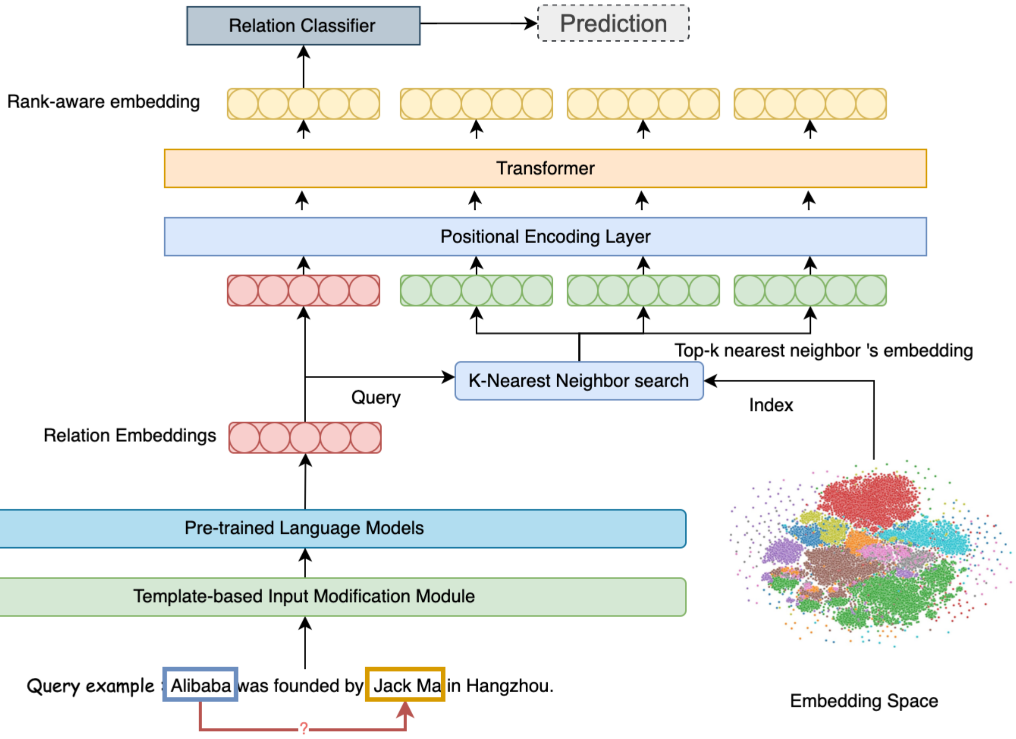

Relation Classifier
Prediction
Rank-aware embedding
Transformer
Positional Encoding Layer
Top-k nearest neighbor 's embedding
K-Nearest Neighbor search
Query
Index
Relation Embeddings
Pre-trained Language Models
Template-based Input Modification Module
Query example : Alibaba was founded by Jack Ma in Hangzhou.
?
Embedding Space

Figure 5.3: The architecture of rank-aware neighbouring encoder.

tion representation $\mathbf{e}_r$ obtained from PLMs will be first merged with the embeddings of its neighbours $\{\mathbf{e}_{n_0}, ..., \mathbf{e}_{n_i}\}$ into the sequence $\mathbf{z} = \{\mathbf{e}_r, \mathbf{e}_{n_0}, ..., \mathbf{e}_{n_i}\}$ in order. As the transformers are neither recurrent nor convolutional, the model architecture restricts its capability of encoding the sequential information in the input. Hence, augmenting the input embeddings with the positional information is crucial for rank-aware learning. We follow the techniques proposed by Vaswani et al. (2017), to get the positional encoding $\mathbf{g}_i$ of each input position i as follows:

$$\omega_k = \frac{1}{10000^{2k/d}}$$

$$\mathbf{g}_i = f(i) = \begin{cases} sin(\omega_k, i), \text{if i = 2k} \\ cos(\omega_k, i), \text{if i = 2k +1} \end{cases}$$

, where d is the encoding dimension. Then, the sequential neighbouring matrix $\mathbf{z}$ is added with the position encoding $\mathbf{g} = \{\mathbf{g}_0, ..., \mathbf{g}_i\}$ into the rank-aware input $\mathbf{x}$. Then, we use the Transformer encoder with a multi-head attention layer and a fully-connected feed-forward layer with ReLU activation to acquire the contextualized encoding of the neighbouring sequence:

$$\mathbf{h} = MultiHead(\mathbf{x})$$
$$\mathbf{y} = FeedForward(\mathbf{h})$$

We take the first contextual embedding which is in the position of the query example from the output encoding $\mathbf{y}$ as the neighbour-aware embedding vector $\mathbf{e}'_r$:

$$\mathbf{e}'_r = \mathbf{y}_0 \tag{5.7}$$

Consequentially, based on the rank-aware neighbouring vector $\mathbf{e}'_r$, we augment the neural relation classifier discussed in Section 5.2.3 with neighbouring attention.

Contrastive Learning with Distant-Peer

Cross-entropy loss, as the most popular loss function for neural classification models in supervised learning, has been long criticized for inducing poor decision margins (Elsayed et al., 2018; Liu et al., 2016) and lacking robustness to noisy annotations (Zhang and Sabuncu, 2018; Sukhbaatar et al., 2015). These two shortcomings are especially fatal for AEC tasks because the target of the neural relation classifier is to learn the correct annotations from the noisy observed annotations. The supervised contrastive loss (Khosla et al., 2020) is an appealing supplement of cross-entropy loss to enhance the noise-tolerant learning for the AEC task.

Contrastive learning has shown signs of resurgence by achieving competitive performance in unsupervised learning in various machine learning domains (Wu et al., 2018; Hénaff, 2020; van den Oord et al., 2018; Hjelm et al., 2019; Chen et al., 2020; He et al., 2020; Stojnic and Risojevic, 2021; Le-Khac et al., 2020). The intuition behind contrastive learning is to pull together an anchor and positive examples meanwhile push apart the anchor from negative examples, where the anchor is the target example. The positive examples are the samples that share the fundamental homogeneity with the anchor example, while the negative examples should be essentially different from the anchor example. Under the unsupervised setting, the positive examples are usually obtained by the data augmentation of the sample, and negative examples are randomly selected from the training batch. Supervised contrastive loss (Khosla et al., 2020) generalizes this idea by introducing the supervising signal to the contrastive process to exploit the label information fully. It pulls closer the normalized embedding from the same class than the embeddings from different classes by taking both the self-augmented examples and the examples with the same label as the positive targets. According to the experimental results presented in Khosla et al. (2020), supervised contrastive loss empirically endows the deep neural networks with better robustness and hyperparameter stability in the context of image corruption and training data reduction regarding Computer Vision (CV).

Contrastive learning also shows its potential advantages in several Natural Language Processing (NLP) tasks, such as text generation (Lee et al., 2021) and information extraction (Ye et al., 2021; Peng et al., 2020). However, controllable self-augmentation is usually harder in most NLP tasks than CV because of the flexibility of natural languages. Therefore, in this project, we merely regard the examples with the same relation label in the batch as positive while taking the example with the different relation label as negative. First, a batch of relation representations $\mathbf{e}_r = \{\mathbf{e}_i\}_{i=1}^{I}$, where $\mathbf{e}_i \in \mathbb{R}^d$, obtained by prompt-based or entity-based approaches are mapped to the dimension reduced projection space p with multi-layer perceptron f_{proj}:

$$\mathbf{z} = f_{proj}(\mathbf{e}_r) \tag{5.8}$$

The vanilla contrastive loss $\mathcal{L}_{cl}$ for the AEC task is:

$$\mathcal{L}_{cl} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{n \in N(i)} exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)} \tag{5.9}$$

, where i is the index of anchor, $P(i)$ and $N(i)$ are the set of indices of all positives and negatives regarding the anchor example in the batch, respectively, $|P(i)|$ is the cardinality of the positive set, and $\tau \in \mathbb{R}^+$ is a scalar temperature hyperparameter. Ideally, the vanilla contrastive loss $\mathcal{L}_{cl}$ lets the relation embedding get closer to the examples from the same class while staying away from the examples from disparate classes. Nevertheless, as the instances in a batch are randomly sampled, the model may not fully leverage the most informative knowledge from the adjoining embeddings of the anchor examples. It is especially problematic when the computational resources limit the batch size, making the sampling less representative and leading to poor classification margins. Therefore, we alleviate this shortcoming by adding neighbouring information to the contrastive loss and involving the supervised signals from cross-entropy learning.

Inserting the positive neighbours to $P(i)$ or negative neighbours to $N(i)$ injects the neighbouring knowledge into the contrastive losses. However, the quality of inserted neighbours strongly impacts the final learning output. Furthermore, there is no guarantee that the observed labels of neighbours are correct. As argued by Lee et al. (2021);

Khosla et al. (2020), the gradient contributions from harder positives or negatives are more conducive to the encoder.

Therefore, instead of randomly choosing the neighbours or inserting all neighbours, we propose a new method to introduce the distant-peer to the positive set for computing contrastive loss. The distant-peer is the farthest positive neighbour according to our defined peer distance λ. The peer distance evaluates if a neighbour is positive or not without considering its own observed labels, but the co-occurrence of its label among all neighbours and the distance from the anchor. Let $\mathbf{L} = \{\tilde{\mathbf{r}}_i\}_{i=0}^{N}$ be the soft-label matrix of neighbours where $\tilde{\mathbf{r}}$ is the KDE-based soft-label of each neighbour computed by Equation 5.6 and N is the number of searched neighbours, and the distance vector of neighbours $\mathbf{D} = \{d_i\}_{i=0}^{N}$ denotes the distances between the anchor and each of its neighbours. Given $\mathbf{L} \in \mathbb{R}^{N\times|\mathcal{R}|}$ and $\mathbf{D} \in \mathbb{R}^{N}$, the peer distance $\lambda \in \mathbb{R}^{N}$ is defined as:

$$\lambda = \mathbf{L}\mathbf{L}^{\intercal} log(\mathbf{D}) \tag{5.10}$$

, where $\mathbf{L}\mathbf{L}^{\intercal} \in \mathbb{R}^{N\times N}$ could be regarded as the co-occurrence of the annotations of neighbours. Multiplying the co-occurrence matrix $\mathbf{L}\mathbf{L}^{\intercal}$ with $log(\mathbf{D})$ embodies our assumption that the dense concentration of concurrent annotations in the distant neighbours may imply the existence of positives. Then, we add the top k neighbours selected by the criterion of peer distance λ to the positive set $P(i)$ in Equation 5.9 to get the neighbour-aware contrastive loss $\mathcal{L}_{ncl}$.

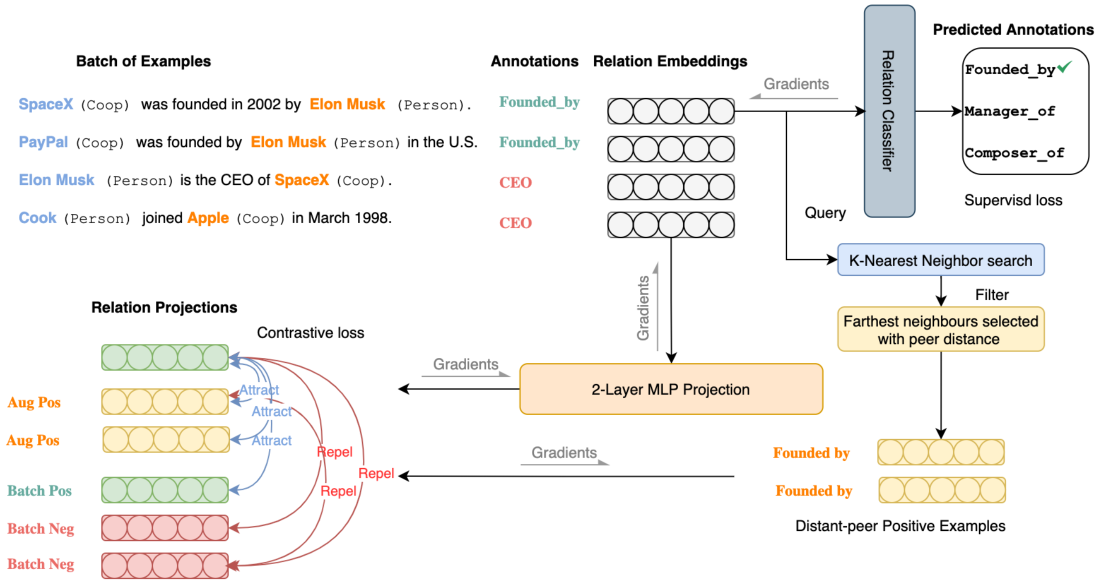


Figure 5.4: The framework of contrastive learning with distant-peer.

Finally, we apply the auxiliary loss technique of multi-task learning (Zhang and

Yang, 2017; Vafaeikia et al., 2020) which combines the strengths of cross-entropy loss and contrastive loss, to incentivize AEC classifier to learn the semantically rich and highly differential relation representations for predicting the corrected labels (Figure 5.4). The standard cross-entropy loss $\mathcal{L}_{ce}$ is computed with the ground-truth revisions and the predictions given by the vanilla softmax-based relation classifier described in Section 5.2.1, while the contrastive loss $\mathcal{L}_{ncl}$ is computed by the distant-peer contrastive learning. We leverage the weighted loss combination $\mathcal{L}$ to introduce the neighbouring knowledge to the neural relation classifier during training:

$$\mathcal{L} = \mathcal{L}_{ce} + \mu \mathcal{L}_{ncl} \tag{5.11}$$

, where μ is a hyper-parameter, the contrastive loss weight.

5.3 Evaluation

We thoroughly evaluate the performance of AEC models with both upstream and downstream methods. Section *Upstream Evaluations*, introduces the classification metrics, accuracy and macro F_1 for upstream evaluation while regarding human revisions as ground truths. Section *Downstream Evaluations*, describes the relation extraction datasets and selected state-of-the-art models for downstream evaluating the de-noising outcomes of our proposed AEC models.

Upstream Evaluations

In essence, the Annotation Error Correction task is the multi-class classification task that requires models to give the revising suggestions as to the same as the human revisers. Hence, naturally, we can take the human revisions in the TACRev and Re-TACRED datasets as the ground truth to evaluate the performance of proposed AEC systems with classification metrics, like accuracy and macro F_1 score (Grishman and Sundheim, 1996). Accuracy can give the general idea of how many predictions are exactly the same as human re-annotations, and F_1 score comprehensively reflect both the correctness and misclassifications. The macro F_1 score is the average of the independent F_1 scores of every class, which is especially conducive to AEC tasks because we expect the AEC models to have evenly performed in rectifying the annotations from all classes. The value of Micro F_1score may be largely increased if the AEC models are good at correcting the major classes, such as the `no_relation` in TACRED, which is misleading for assessing the model performance of dealing with the annotation noise from minor classes.

Downstream Evaluations

While the classification metrics are simple and straightforward approaches to quantifying the performance of the AEC models, they may not adequately represent the real-world performance of proposed systems directly. Hence, downstream evaluation with the state-of-the-art (SOTA) models in relation extraction may present more convincing evaluation results: (1) Using the optimized AEC systems to automatically denoise the original Train set of TACRED (Zhang et al., 2017a) and DocRED dataset (Yao et al., 2019b); (2) Training the same SOTA relation extraction models on both the raw Train set and the denoised Train set; (3) Evaluating the models trained on different Train sets with the same Dev and Test sets on TACRED, TACRev (Alt et al., 2020), and DocRED (Yao et al., 2019b).

Table 5.1 shows that both TACRED and TACRev share the same size Train, Dev, and Test set with 68,124 examples, 22,631 examples, and 15,509 examples, respectively. While the Train sets of TACRED and TACRev are identical, TACRev has 1,656 examples in its Dev set and 998 examples in its Test set that are revised from the original TACRED Dev and Test examples. The authors of TACRev indicate that the evaluation quality is largely improved by these 7.3% and 6.4% label revisions on TACRED Dev and Test sets because the erroneous labels contribute up to 8% test error. The best AEC model selected by the upstream classification metrics changes the annotations of 7,267 examples in the TACRED Train set, which accounts for 10.7% of the Train examples. The DocRED datasets include 38,269 Train examples, 12,332 Dev examples, and 12,842 Test examples. The optimal AEC models automatically revise 7,676 examples on the Train set, accounting for 20.0% Train instances.

To evaluate our proposed AEC models on downstream performance we first select the SOTA sentence-level relation extraction model by Zhou and Chen (2021) and the SOTA document-level ATLOP models (Zhou et al., 2020). For instance, as for sentence-level models, we first train a model on the original Train set of TACRED and another model on an automatically denoised Train set of TACRED. Finally, by comparing the performance differences between these two trained models on the same original Test set of TACRED, we can testify whether correcting the training set by AEC models is conducive to the downstream tasks.

Dataset	#Train	#Dev	#Test	#AEC Revisions	#RelTypes
Sentence-level Relation Extraction Datasets					
TACRED (Zhang et al., 2017a)	68,124	22,631	15,509	7,267	42
TACRev (Alt et al., 2020)	68,124	22,631	15,509	7,267	42
Document-level Relation Extraction Datasets					
DocRED (Yao et al., 2019b)	38,269	12,332	12,842	7,676	96

Table 5.1: The statistics of the original TACRED, DocRED datasets and the TACRev dataset partially revised from TACRED. Our proposed AEC models revised 10.7% examples on the TACRED Train set, and 20.0% examples on DocRED Train set.

Details regarding these two investigated models are as follows:

Entity Marker Model The entity marker model proposed by Zhou and Chen (2021), has almost the same architecture as the vanilla PLM-based relation classifiers described in Sections 5.2.1, with the entity-based relation embedder applying the ENTITY MARKER techniques demonstrated in Section 4.2.2. This simple but strong baseline offers the new SOTA performance in the sentence-level RE task. It even outperforms the competitive knowledge-enhanced PLM KnowBERT (Peters et al., 2019) on TACRED.

ATLOP model ATLOP, Adaptive Thresholding and Localized Context Pooling, model presented by Zhou et al. (2020) tackles the document-level RE tasks with two techniques:

- Adaptive-thresholding loss: Most RE models on DocRED apply a threshold to the output probability for deciding if a certain relation holds between given head and tail entities. But the threshold needs to be manually specified and so can potentially result in decision errors. Adaptive-thresholding enables the model to learn an adaptive threshold independently, depending on entity pairs.
- Localized context pooling: Normally, the entity representations for document-level RE tasks are acquired by aggregating the embeddings of all occurrences for a given entity over entire documents, whereas some context of the entities may not be relevant and may even distract from the RE targets. Hence, the localized context pooling reinforced the capability of capturing related context for entity pairs of PLMs by transferring pre-trained attention, which leads to better entity representations.

The ATLOP model reaches an F_1 score of 63.4 on DocRED, surpassing other representative document-level RE models such as HIN-BERT-base (Tang et al., 2020) and CorefBERT-base (Ye et al., 2020).

5.4 Experiments

- Section 5.4.1 gives the detailed description of experimental implementations in AEC.
- Section 5.4.2 contrasts the fine-tuned relation representations introduced in Section 5.2.1 with the zero-shot relation representations previously discussed in Section 4.2.2 in AEC through comprehensive experiments.
- Section 5.4.3 presents the results of different types of uncertain labels introduced in Section 5.2.2.
- Section 5.4.4 demonstrates the performance of neighbour-aware AEC models discussed in Section 5.2.3.
- Section 5.4.5 shows the impressive performance gain with the combination of uncertain labels (Section 5.2.2) and distant-peer contrastive loss (Section 5.2.3).
- Section 5.4.6 reveals the downstream performance of our proposed AEC models following the descriptions in Section 5.3.

5.4.1 Implementation Details

The Pre-trained Language Model, long sequence handling techniques and experimental environments for the AEC task are the same as the AID task described in Section 4.4.1. The reported experimental results on the Test sets in this section are regarding the models that reach the highest macro F_1 scores on corresponding Dev sets within the training epochs.

Datasets The proposed uncertain labeling methods and neighbour-aware classifiers rely on neighbouring information by retrieving the neighbours for the query example. Identically to the development of the datastore used in AID task, the key of embeddings and value of labels are initially acquired based on the entire TACRED dataset with 103,738 examples and DocRED dataset with 50,503 examples. The upstream experiments are conducted with the TACRev, Re-TACRED and Re-DocRED datasets which contain manual revisions of annotations of partial examples in TACRED and DocRED. We take the revised labels as the gold labels to evaluate the performance of proposed AEC methods. The TACRev dataset contains 1,263 Dev examples and 1,263 Test examples, Re-TACRED contains 5,364 Dev examples and 5,365 Test examples, and Re-TACRED contains 206 Test examples and 205 Dev examples. As for the downstream evaluations, we leverage the initial TACRED and DocRED with the original data split depicted in Table 5.1 to get the comparable results with other existing research on relation extraction.

Embedding of Neighbours The parameters of the PLM are fine-tuned along with the iterations of cross-validation. This means that the initial relation embedding acquired by the off-shelf PLM stored in the keybase for the KNN retriever may be out-of-date. Therefore, we investigate the influence of the static and dynamic neighbouring embeddings:

- **Static Embedding of Neighbours**: The embeddings in the keybase always keep the same initialization from the off-shelf PLM.
- **Dynamic Embedding of Neighbours**: The embeddings in the keybase are updated at the end of each epoch with the new embeddings acquired by latest fine-tuned models.

Empirically, we found that the rank-aware neighbouring encoders work better with static neighbour embeddings, whereas the contrastive learning approaches work better with the dynamic neighbour embeddings.

Hyper-parameters of Cross Validation Both TACRED and DocRED are split into 4 folds for cross validation, and each iteration of cross validation has 5 training epochs.The optimizer for all types of neighbour-aware classifiers is AdamW (Loshchilov and Hutter, 2019). The learning rate for fine-tuning the AEC models is 5e-4, the dropout probability is 0.2, and the warm-up ratio is 0.1 for both datasets.

Hyper-parameters of Uncertain Labelling Models With the KNN-based label replacement methods, we only study the setting of $k = 1$. With Bayesian optimization (Snoek et al., 2012) on the Dev sets, we optimize the replacement threshold ϕ of KNN-based methods in Equation 5.5 as 0.3 for TACRED and 0.15 for DocRED, and the bandwidth h for KDE-based soft labelling in Equation 5.6 as 0.25 for TACRED and 0.1 for DocRED.

Hyper-parameters of Neighbour-aware Classifiers

- **Rank-aware neighbouring Encoder**: The backbone model is the Transformer encoder with 6 layers where each attention layer has 8 attention heads. The dropout probability of each layer is 0.1 as default. The number of neighbours retrieved for providing contextual information is 10.
- **Contrastive Learning with Distant-Peer**: The dimension of the projection space for conducting contrastive learning is 189, which is one fourth of the embedding dimension of `bert-base-cased` models. In total, 100 neighbours are retrieved with their peer distance computed according to Equation 5.10, and only 5 of them are selected as the positives that would be involved into the contrastive learning. The contrastive learning weight μ is 0.35 for TACRED and 0.02 for DocRED. The number of selected distant-peers and the contrastive learning weight μ are tuned with Bayesian optimization (Snoek et al., 2012) on the Dev sets.

Hyper-parameters of Downstream Models The hyper-parameters of the downstream models on TACRED are identical to the original settings reported by Zhou and Chen (2021), regardless of whether the models are trained on the original Train set or the denoised Train set. As for ATLOP model on DocRED, we modified the learning rate from its original configured 5e-5 into 2e-5 for `roberta-large` based models. The warm-up ratio is set as 0.36 for `roberta-large` based models and 0.04 for `bert-base-cased` based models.

5.4.2 Zero-shot KNN vs. Fine-tuned Neural Corrector

According to the experimental results in Table 5.2, the PLM-based neural classifiers fine-tuned by cross-validation show a large advantage in AEC tasks compared to zero-shot KNN classifiers. It demonstrates that fine-tuning the PLM with the task of relation extraction is conducive to correcting annotations, because of mitigating the ambiguity of the AEC tasks for the PLMs. The neural classifiers reach the highest macro F_1 scores of 64.4 on TACRev, of 46.4 on Re-TACRED, and of 44.5 on Re-DocRED. Therefore, we regard the vanilla PLM-based neural classifiers as a strong baseline for contrasting our further amelioration in AEC learning.

		TACRev		Re-TACRED		Re-DocRED	
Detection Methods	Relation Representation	Acc	F_1	Acc	F_1	Acc	F_1
KNN Classifier (k=1) +Static Representations	Relation Prompt	56.4	39.8	39.0	31.0	47.8	34.9
	Entity marker	54.2	38.5	36.8	26.5	44.4	33.1
	Entity marker (punct)	56.6	39.5	36.8	27.3	43.9	30.9
KNN Classifier (k=3) +Static Representations	Relation Prompt	45.9	26.0	27.9	18.4	50.7	36.1
	Entity marker	46.1	25.4	26.4	13.6	44.9	29.8
	Entity marker (punct)	49.1	25.5	27.1	16.5	43.4	29.2
Neural Classifier Dynamic Representations	Relation Prompt	75.2	60.6	49.4	44.4	**57.3**	**44.5**
	Entity marker	**75.7**	**64.4**	49.7	**46.4**	52.4	28.7
	Entity marker (punct)	72.6	59.1	**50.8**	44.5	55.3	32.7

Table 5.2: The accuracy and macro F_1 of the KNN classifiers with static representations and neural classifiers with the dynamic representations on the Test set of TACRev, Re-TACRED and Re-DocRED. The PLM-based neural classifiers fine-tuned by cross-validation consistently outperform the static methods.

Among the KNN classifiers, we found that the models with $k = 1$ slightly outperform the models with $k = 3$ on TACRev and Re-TACRED, which differs from the observations in AID tasks (Section 4.4.3). As for the AID tasks, we reckon that having more voting neighbours helps detect abnormal annotations on TACRED because more local geometry information in the embedding space is available. However, the empirical results indicate that the successful methods in AID are not necessarily effective in AEC equally. According to our analysis, the `no_relation` examples are pervasive in TACRED. Therefore those `no_relation` neighbours that are considered helpful in AID may mislead the KNN-based classifier in AEC. In contrast, the KNN classifiers with $k = 3$ still outperform those with $k = 1$ on DocRED if without distracting `no_relation` examples.

As for the KNN-based classifiers, the RELATION PROMPT embedders consistently outperform the ENTITY MARKER and ENTITY MARKER (PUNCT) embedders, regardless of $k = 1$ or $k = 3$, which supports the implications in AID task. Nevertheless, this tendency is not held under the circumstance of cross-validation. Since fine-tuning with cross-validation lets models learn the task-specific contextualized embeddings for the newly inserted tokens needed for ENTITY MARKER and ENTITY MARKER (PUNCT) representations, they show competitive abilities on TACRev and Re-TACRED. The ENTITY MARKER based neural classifiers even increases the macro F_1 by 3.8 on TACRev, and by 2.0 on Re-TACRED, compared to RELATION PROMPT based methods. However, cross validated on Re-DocRED, the RELATION PROMPT representations show overwhelming improvement over ENTITY MARKER and ENTITY MARKER (PUNCT) representations, with a macro F_1 gain of 15.8 and 11.8 respectively. It is also noticeable that different entity marker types, namely new special tokens or punctuation, have their own advantages and shortcomings after cross-validation learning.

5.4.3 Uncertain Labeling

Furthermore, as shown in Table 5.3, the uncertain labels proposed in Section 5.4.3 potentially enable neural classifier to better correcting suspicious annotations than certain labels. Empirically, we find that the models with ENTITY MARKER representations are less sensitive to the certainty of learning objectives, than the ENTITY MARKER (PUNCT) and RELATION PROMPT representations. For instance, for the models with ENTITY MARKER (PUNCT) representations, the KNN-based uncertain labels grant the increase of macro F_1 of 2.6 on TACRev, 1.8 on Re-TACRED, and 1.7 on Re-DocRED, but merely 0.1 on TACRev, 0 on Re-TACRED, and 0.4 on Re-TACRED for those with ENTITY MARKER representations. Similar phenomena can be observed with the approach of KDE-based soft labels, where the uncertain labels even compromise the performance of the model with ENTITY MARKER representations by decreasing macro F_1 by 0.4. We suspect that it is caused by properties of the contextualized embeddings of the newly introduced tokens after fine-tuning by cross-validation, which will be left as future work.

Generally, the KDE-based approaches of deducing the uncertain labels provide more benefits for the AEC models than KNN-based methods. Taking the models based on RELATION PROMPT as examples, the KDE methods improve the macro F_1 by 5.9 % on TACRev, 4.2 % on Re-TACRED and 18.8 % on Re-DocRED. The KNN-based label replacement only improves macro F_1 by 1.8 % on TACRev, 1.1 % on Re-TACRED, but result in a decrease of 0.1% on Re-DocRED.

5.4.4 Neighbour-aware Classifiers

Considering the results in Table 5.4, we would argue that the distant-peer contrastive learning is a better mechanism to inject the neighbouring knowledge into the AEC classifiers than the rank-aware neighbouring encoder. As observed, the rank-aware neighbouring encoder only improves the macro F_1 of the AEC models with the ENTITY MARKER (PUNCT) on TACRev, ENTITY MARKER (PUNCT) on both Re-TACRED and

		TACRev		Re-TACRED		Re-DocRED	
Labeling Methods	Relation Representation	Acc	F_1	Acc	F_1	Acc	F_1
Certain Label	Relation Prompt	75.2	60.6	49.4	44.4	57.3	44.5
	Entity marker	75.7	64.4	49.7	46.4	52.4	28.7
	Entity marker (punct)	72.6	59.1	50.8	44.5	55.3	32.7
KNN Uncertain Label	Relation Prompt	70.8	61.7	49.6	44.9	57.3	43.9
	Entity marker	**75.8**	**64.5**	49.7	46.4	50.5	29.1
	Entity marker (punct)	71.1	61.7	50.6	45.8	53.4	34.4
KDE Uncertain Label	Relation Prompt	71.1	64.2	50.2	46.3	**65.0**	**52.9**
	Entity marker	74.5	64.0	**51.4**	**46.9**	52.4	29.9
	Entity marker (punct)	73.6	62.7	49.4	45.2	58.3	40.5

Table 5.3: The accuracy and macro F_1 of the vanilla AEC models learnt with the labels with different certainty on the Test set of TACRev, Re-TACRED and Re-DocRED. The labels with uncertainty likely contribute to better performance in correcting annotations than overconfident hard labels.

Re-DocRED. Apart from these three settings, the rank-aware neighbouring encoders at most time disappointedly worse the performance of AEC models, especially for Re-DocRED. On the opposite, the distant-peer contrastive learning is considered to have the desired property of aiding the representation learning for tackling the AEC task. Aside from slightly perturbing the performance of the AEC models with RELATION PROMPT on TACRev and Re-TACRED, the multi-task loss combing the distant-peer contrastive loss and cross-entropy loss guides the models to correcting the annotation errors with enhanced prudence.

Regardless of the disillusionary overall performance of rank-aware neighbouring encoder, it reaches the highest macro F_1 on Re-DocRED with the RELATION PROMPT representations, exceeding the baseline by 9.2% and surpassing the contrastive models by 11.7%. This unexpected finding arouses the suspicion that the contrastive methods may be more vulnerable to the overconfident labels, which are supported by the results presented in the next section.

5.4.5 Contrastive Learning with Uncertainty

After a comprehensive search for the best setup to combine the strengths of the uncertain labelling and neighbour-aware classifiers, we found that the distant-peer contrastive models learnt with KDE-based soft labels always lead to the best results among

		TACRev		Re-TACRED		Re-DocRED	
Labeling Methods	Relation Representation	Acc	F_1	Acc	F_1	Acc	F_1
Neural Classifier	Relation Prompt	75.2	60.6	49.4	44.4	57.3	44.5
	Entity marker	**75.7**	64.4	49.7	46.4	52.4	28.7
	Entity marker (punct)	72.6	59.1	**50.8**	44.5	55.3	32.7
+ Neighbouring Encoder	Relation Prompt	70.1	58.8	49.7	44.8	**59.2**	**48.6**
	Entity marker	69.4	61.3	49.0	45.8	37.9	19.7
	Entity marker (punct)	67.6	60.3	47.5	43.0	33.0	17.4
+ Contrastive Learning	Relation Prompt	70.3	60.3	50.3	45.5	58.3	43.5
	Entity marker	74.0	64.4	50.2	**47.0**	56.3	37.5
	Entity marker (punct)	72.8	**65.1**	49.8	46.9	55.3	34.2

Table 5.4: The accuracy and macro F_1 of different approaches that endow the neural relation classifiers with the neighbouring awareness with the certain labels on the Test set of TACRev, Re-TACRED and Re-DocRED. The neighbouring encoding and contrastive learning potentially make the classier to perform better at rectifying the annotations.

all configurations (Table 5.5). The increment of macro F_1 is decomposed by ablating the contribution made by the uncertain labels and neighbour-awareness.

On TACRev, solely applying the KDE uncertain labels decrease the macro F_1 by 0.3, but combing soft labels with distant-peer contrastive learning leads to the highest accuracy of 75.8 and macro $F_1$of 66.2. Similarly, on Re-DocRED, independent usage of distant-peer contrastive learning reduces the macro F_1 by 1.0, whereas collaborating with the KDE-based soft labels result in a significant improvement that reaches the highest accuracy of 66.0 and macro F_1 of 57.8.

The observations reveal two implications: (1) Although the models with ENTITY MARKER representations already show considerable robustness in dealing with the overconfident labels, properly introducing the uncertainty is still meaningful if the models are further ameliorated with distant-peer contrastive learning. (2) The distant-peer contrastive learning is sensitive to the noise in certain labels when the models are based on RELATION PROMPT representations, but the KDE-based soft labelling can mitigate the vulnerability to a large extent.

The statistics in Table 5.1 implies that the optimized AEC models would totally revise 10.7% examples on the TACRED Train set, and 20.0% examples on DocRED Train set. Furthermore, we visualize a part of decisions made by our proposed AEC models to have a more intuitive sense of the consequence. Figure 5.5 depicts the partial revising outcome by the optimal models on TACRED. The examples were initially la-

Methods	Acc	F_1	ΔF_1
Best Combination on TACRev			
Neural Classifier with Entity Marker Representation	75.6	64.3	
+ KDE Uncertain Label	74.5	64.0	- 0.3
+ Distant-peer Contrastive Learning	74.0	64.5	+ 0.2
+ KDE Uncertain Label	**75.8**	**66.2**	**+ 1.9**
Best Combination on Re-TACRED			
Neural Classifier with Entity Marker Representation	49.3	44.3	
+ KDE Uncertain Label	51.4	46.9	+ 2.6
+ Distant-peer Contrastive Learning	50.2	47.0	+ 2.7
+ KDE Uncertain Label	**52.2**	**47.7**	**+ 3.4**
Best Combination on Re-DocRED			
Neural Classifier with Relation Prompt Representation	57.2	44.5	
+ KDE Uncertain Label	65.0	52.9	+ 8.4
+ Distant-peer Contrastive Learning	58.2	43.5	-1.0
+ KDE Uncertain Label	**66.0**	**57.8**	**+ 13.3**

Table 5.5: The macro F_1 and the improvement of the macro F_1 under the best combinations of proposed methods on different datasets. The combination of KDE uncertain labels and distant-peer contrastive learning evidently unleashes the true potential of AEC models.

beled with the relations `per:title`, `per:employee_of` and `per:top_members/employees`, that are ambiguous and error-prone to annotate as discussed in Section 4.5.1. After automatic rectification, most examples have been re-labeled as `no_relation`, which is compatible with the propensity of revisions made in TACRev. The examples initially labeled as `per:top_members/employees` (a red point) with the representation located in the cluster of `per:employee_of` (brown points) is revised into `per:employee_of` (brown points) by the AEC models. Several `per:top_members/employees` labels are reasonably changed into `org:founded_by`, `org:political/religious_affiliation` or `per:children`. Overall, the automatic revisions are congenial with reason and common sense.

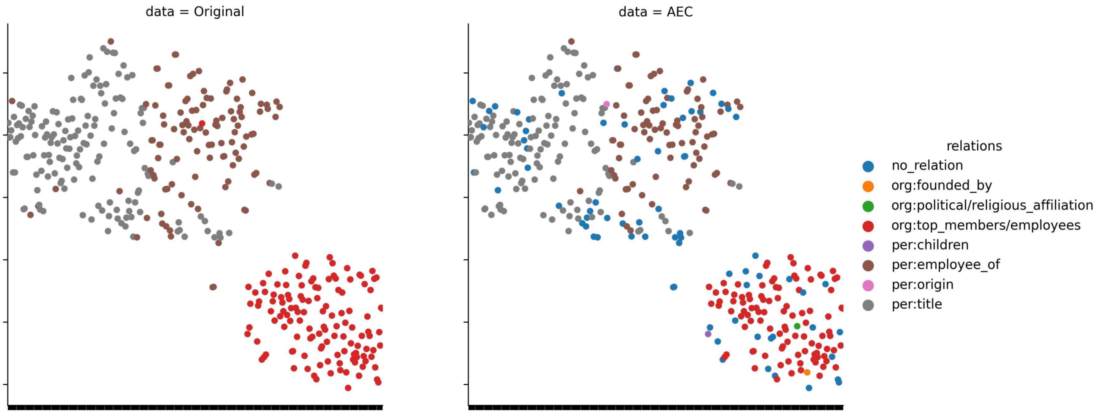


Figure 5.5: Illustration of the revising outcome on TACRED by the optimal AEC models, presented by visualizing the RELATION PROMPT embedding of randomly sampled instances from relations `per:title`, `per:employee_of` and `per:top_members/employees` classes. **Left**: examples with the original annotations. **Right**: examples with the revised annotations predicted by the AEC models.

5.4.6 Learning on Denoised Train Sets

Assuming the annotation noise is evenly spread across all data splits, the test error caused by the noise in Test sets may not fully reflect the true contribution of AEC. Still, training on the Train set denoised by our proposed AEC model empirically led to the enhancement of both sentence-level and document-level relation extraction models, even evaluated on the probably flawed Test sets of TACRED, TACRev and DocRED. According to Yao et al. (2019b), the Ign F_1 reported on. DocRED is the Micro F_1 excluding those relational facts shared by the Train, Dev and Test sets.

TACRED

In Table 5.6, we recognize the persistent advancement by training the models with ENTITY MARKER representations on denoised TACRED Train set. Especially, the model based on BERT-base receives the maximum gain of Micro F_1 by 3.5 % on TACRED and by 3.4 % on TACRev by learning with denoised annotations. The model based on RoBERTa-large initially is inferior to the KnowBERT (Peters et al., 2019) when the predictions are evaluated with TACRED Test sets. However, the enhancement brought by the denoised data can be captured by evaluating with both the Test sets of TACRED and TACRev, enabling the model to reach higher Micro F_1 on TACRED Test set than the KnowBERT (Peters et al., 2019).

Models	TACRED Micro F_1	TACRev Micro F_1
Other Models Trained on TACRED Train Set		
PA-LSTM (Zhang et al., 2017b)	65.1	73.3
C-GCN (Zhang et al., 2018)	66.3	74.6
SpanBERT(Joshi et al., 2020)	70.8	78.0
KnowBERT (Peters et al., 2019)	71.5	79.3
Investigated Models Trained on TACRED Train Set		
BERT-base + entity marker (Zhou and Chen, 2021)	68.4	77.2
BERT-large + entity marker (Zhou and Chen, 2021)	69.7	77.9
RoBERTa-large + entity marker (Zhou and Chen, 2021)	70.7	81.2
Investigated Models Trained on De-noised TACRED Train Set		
BERT-base + entity marker	70.8	79.8
BERT-large + entity marker	71.1	80.4
RoBERTa-large + entity marker	**72.1**	**81.9**

Table 5.6: The Micro F_1 Test scores on TACRED and TACRev of the sentence-level relation extraction models trained on original Train set and automatically denoised Train set. Our proposed AEC models can further enhance the SOTA models by improving the quality of training data.

DocRED

The results in Table 5.7 demonstrate that the denoised data are also beneficial to the SOTA document-level relation extraction models. However, the advantage of denoised data is less obvious than we observed with the investigated models on TACRED. Automatically denoised data slightly increases the Ign F_1 and F_1 of ATLOP BERT-base model by 1.1% and 0.7% respectively, and improve the Ign F_1 and F_1 of ATLOP RoBERTa-large model by 0.6% and 0.8%. Therefore, it is reasonable to believe that our currently implemented AEC model may be more successful in correcting the relation annotation in the sentence-level than document-level. Since document-level relation extraction task usually makes greater demands of capturing the inter-sentence interactions, adding hierarchical context modelling to AEC models is a promising direction for future exploration.

Models	DocRED	
	Ign F_1	F_1
Other Models Trained on DocRED Train Set		
CNN (Yao et al., 2019b)	40.33	42.26
BiLSTM (Yao et al., 2019b)	48.78	51.06
BERT-LSR-base (Nan et al., 2020)	56.97	59.05
HIN-BERT-base (Tang et al., 2020)	53.70	55.60
CorefBERT-base (Ye et al., 2020)	54.54	56.96
CorefRoBERTa-large (Ye et al., 2020)	57.90	60.25
Investigated Models Trained on DocRED Train Set		
ATLOP-BERT-base (Zhou et al., 2020)	59.31	61.30
ATLOP-RoBERTa-large (Zhou et al., 2020)	61.39	63.40
Investigated Models Trained on De-noised DocRED Train Set		
ATLOP-BERT-base	59.98	61.73
ATLOP-RoBERTa-large	**61.75**	**63.88**

Table 5.7: The Micro F_1 and Ign Micro F_1 Test scores on DocRED of the document-level relation extraction models trained with the original Train examples and the denoised Train examples. Similarly, automatically rectifying the Train set leads to the boost of performance in document-level relation extraction.

5.5 Discussions

5.5.1 Revision Quality of Automatic Re-Annotator

We further analyze the revision quality of our proposed AEC methods from two aspects: (1) quantification of the agreement between automatic revisions and manual revisions, and (2) case analysis of the positive and negative predictions by AEC models.

Agreement between Human and AEC Model

Cohen Kappa score κ (Cohen, 1960; Artstein and Poesio, 2008) is a widely accepted method to assess the rate of agreement between two different annotators, which is also informative to quantify the consistency between human and machine revisions. Compared to the F_1 score we used for evaluating the AEC performance while regarding the human revision as the ground truth, the Cohen Kappa score tries to contrast the observed agreement between the human and AEC model with the expected agreement when both of them revise labels randomly. The range of Cohen Kappa score is $[-1, 1]$, and the score above 0 indicates that there is an agreement between two raters, and the score above 0.8 generally means the agreement is considerable.

The Cohen Kappa score κ reported in Table 5.8 indicates that: (1) The annotators composing TACRev and Re-TACRED are in excellent agreement and awarded with the Cohen Kappa score of 0.882. (2) The Cohen Kappa score between TACRev and the revisions by the AEC model is 0.587. It reveals that though the agreement between humans and the AEC model is worse than inter-human, the decisions made by the AEC model still demonstrate good consistency with human revisions. Generally, on the basis of the Cohen Kappa score κ, the overall agreement between humans and our proposed AEC model is acceptable.

Dataset	Revision A	Revision B	Cohen Kappa κ
TACRED	TACRev	Re-TACRED	0.882
	TACRev	AEC TACRED	0.587
DocRED	Re-DocRED	AEC DocRED	0.604

Table 5.8: The Cohen kappa score between different revisions on TACRED and DocRED. The corrections predicted by the optimized AEC models demonstrate promising agreement with the manual revisions.

Case Study of Automatic Revision

To better understand the actual performance of our proposed AEC models, we manually re-examine around 100 revising decisions made on TACRED and DocRED, and the typical positive and negative corrections are listed in Table 5.9.

The positive example in TACRED was originally annotated as `per:other family`, but the word ”daughter” in the context strongly hints that `per:children` may be a better label in this case, which has been nicely apprehended by our proposed model. The context of exemplified negative example in TACRED is relatively intricate even for the human annotators because it would easily mislead us to believe that ”City Council” is the workplace of ”Dixon”. However, if we read carefully, we could notice that this assumption is actually incorrect in the given context, so `no_relation` would be a more appropriate label here. Although the AEC model also could not give the correct answer, letting the relation label remain the same as the original one is a rational choice that would not introduce more noise after automatic revision.

The head and tail entities of the positive example in DocRED are the famous English writer ”John Fowles” and one of his well-known novels ”The French Lieutenant’s Woman”. Possibly due to the distracting word ”French” in the book title, its original relation label in DocRED is `country`, which is incorrect. Conversely, our proposed AEC model reasonably revised the label into `notable work` in agreement with human revisers, which implicitly reflects that pre-training on the large-scale general corpus like Wikipedia educates the PLMs with common-sense knowledge. As for the negative example in DocRED, the originally labelled relation between ”Thailand” and ”Thai” is `located in`. This is incorrect because the word ”Thai” can only refer to a native or inhabitant of Thailand or to the official language of Thailand. In contrast, the AEC revision of `ethnic group` is more understandable compared to the original annotation, though it may not be as accurate as of the human revision of `official language` in the given context.

Examples of Automatic Corrections on TACRED				
Pos	**[Tessa Dahl]** ’s daughter is the model and writer <**Sophie Dahl**>.	**Annotation:** `per:other_family`	**AEC Model:** `per:children`	**Human:** `per:children`
Neg	If those efforts fail, **[Dixon]** would probably be forced from office and the <**City Council**> president, Stephanie Rawlings-Blake, would succeed her.	**Annotation:** `per:employee_of`	**AEC Model:** `per:employee_of`	**Human:** `no_relation`
Examples of Automatic Corrections on DocRED				
Pos	Works often described as examples of historiographical metafiction include: William Shakespeare ’s Pericles, Prince of Tyre (c.1608), **[John Fowles]**’s <**The French Lieutenant ’s Woman**> (1969), ...	**Annotation:** `country`	**AEC Model:** `notable work`	**Human:** `notable work`
Neg	Chupong Changprung (RTGS: Dan Chupong); born March 23, 1981 in Kalasin Province, **[Thailand]**, <**Thai**> nickname: ”Deaw” is a <**Thai**> martial arts film actor	**Annotation:** `located in`	**AEC Model:** `ethnic group`	**Human:** `official language`

Table 5.9: The case study of positive and negative annotation corrections made by our proposed AEC models. The models have the due capability in making the reasonable decision of rectifying annotations, though they may not be as precise as human revisions.

5.5.2 Efficiency of Automatic Re-Annotator

Table 5.10 shows the statistics of the average time in seconds required for human and automatic re-annotators to reexamine and possibly revise each sample of TACRED and DocRED, respectively. We estimate the efficiency of human revisers through the annotation progress of the re-annotator who helped us revise 411 examples of DocRED in 35 hours. A more detailed analysis of the efficiency of human re-annotators will be conducted in the future. The annotation speed of the annotation inconsistency detector and annotation error corrector is computed by the averaging run time of all experiments, including different configurations. Although the estimated efficiency may not be precise, we still can compare performance in terms of orders of magnitude.

Re-Annotator	TACRED	DocRED
Human Reviser	47.68	306.63
Annotation Inconsistency Detector	4.27×10^{-3}	6.69×10^{-3}
Annotation Error Corrector	0.32	3.33

Table 5.10: Comparison the efficiency between human revisers and our proposed two types of automatic re-annotators, by the average time (in seconds) of relabeling each example. The PLM-based re-annotators are at least hundreds of times faster than the human revisers.

On average, the human revisers need to spend around 47.68 seconds to revise each annotation on sentence-level relation extraction corpus TACRED and 306.63 seconds, ($\sim$5 minutes), to revise each annotation on document-level corpus DocRED. In comparison, our proposed automatic re-annotators are significantly faster. As the annotation inconsistency detectors only involve the inference stage of PLMs, they only take 4.27×10^{-3} seconds and 6.69×10^{-3} seconds to assess the validity of each annotation on TACRED and DocRED, respectively, which is over 10^4 times faster than the human. In contrast, the annotation error corrector depends on the time-consuming fine-tuning stage of PLMs to give exact suggestions in revising annotations by cross-validation. It would cost about 0.32 seconds to revise on TACRED and 3.33 seconds on DocRED. The difference between re-annotation efficiency on TACRED and DocRED is due to the difference in their input length. Despite the annotation error corrector being slower than annotation inconsistency detectors, it still demonstrates the re-annotation speed at least hundreds of times higher than the human re-annotators.

In sum, our proposed pair of re-annotators could improve the effective efficiency of re-annotation. The annotation inconsistency detectors can help spot the most suspicious labels for human revisers, in order to reduce the overall time for dataset revision or filter out the unreliable samples from the training set to alleviate the negative effects. Moreover, the annotation error corrector based on the prior knowledge in PLMs shows both competitive re-annotation efficiency and accuracy, even when compared with human revisers.

Chapter 6

Conclusions and Future Directions

The re-annotation tasks we investigated, Annotation Inconsistency Detection and Annotation Error Correction, have significant implications for prospective data-driven Natural Language Processing researches. Through empirical study, we demonstrate the non-negligible potential of Pre-trained Language Models in reducing annotation noise. Moreover, we point out the appealing directions to further extend our work in the future.

6.1 Conclusions

In conclusion, Pre-trained Language Models (PLMs) show impressive potential as automatic re-annotators. Through detailed experiments conducted on sentence-level and document-level relation extraction datasets (TACRED and DocRED) and their human revised versions, we have shown that a PLM-based annotation inconsistency detector and annotation error corrector can be used to improve annotation quality more efficiently.

Our investigations in Annotation Inconsistency Detection indicate a well-designed prompt can induce PLMs to acquire instance representations with favourable properties to detect inconsistencies in zero-shot scenarios, which profits from prior knowledge in PLMs. Compared to the K-Nearest Neighbour classifier, our proposed credibility scores jointly consider both the distance and trustworthiness of K-Nearest neighbours. The credibility-based detector yields better sensitivity and specificity of detecting annotation inconsistencies than the vote-based K-Nearest Neighbour classifier. The best binary F_1 scores that our proposed inconsistency detector can reach are 92.4 on TACRED and 72.5 on DocRED.

Our observations in Annotation Error Correction suggest that fine-tuning with proper domain-specific knowledge by cross-validation further improves the capability of PLMs in recommending precise revisions for the suspicious annotations. Given the vulnerability of the softmax classifier to annotation noise, we mitigate the overconfidence of the automatic corrector by introducing uncertainty to observed hard labels. We combine a novel distant-peer contrastive loss with the ordinary cross-entropy loss to improve neighbour awareness while learning to correct annotations. The distant-peer contrastive framework selects the positive neighbours by their label co-occurrence and distance from the anchor example. The results demonstrate that the distant-peer contrastive corrector learnt with Kernel Density Estimation based soft label obtains the best performance in annotation correction throughout the study. The highest macro F_1 scores we observed throughout the study of error corrector are 66.2 on TACRED and 57.8 on DocRED.

In practice, we prove that the state-of-the-art model can benefit from learning using data automatically denoised by our proposed annotation corrector. Automatic correction at most leads to a gain of 3.6% to the downstream state-of-the-art relation extraction models. In terms of time cost, an automatic inconsistency detector is capable of reexamining each sample in the dataset over ten thousand times more efficiently than a human. Moreover, an automatic error corrector can even suggest the recommended revision for annotations in question hundreds of times faster compared to manual revision with acceptable accuracy. We believe that our findings are extremely valuable to prospective data-driven NLP research.

6.2 Future Directions

This dissertation project opens up the following potential directions for future research in data-driven NLP:

- For example, we did not explore re-annotating very domain-specific datasets, such as medical reports or legal documents. Annotation noise is also assumed to be pervasive in those datasets, so it would be of interest to investigate if domain-specific PLMs (Lee et al., 2020; Feng et al., 2020; Li et al., 2020b) can help to reveal the annotation inconsistencies and errors in such domains.

- For the credibility score, we manually tuned the threshold for the credibility-based detectors, but learning the threshold automatically would be worth exploring as well.

- Through our experiments contrasting the influence of different input formats to the annotation corrector, we surprisingly found that the `Entity Marker` with newly introduced special tokens show outstanding robustness to training noise. Considering there is no systematic analysis on the effect of freshly introduced special tokens, we are also curious about their potential advantages in noise-tolerant learning.

- Sentence-level and document-level annotation revisions are treated similarly by the automatic re-annotators presented in this dissertation, without explicit consideration of complex inter-sentence interactions and document structure. Therefore, we believe that methods for modelling the document hierarchy or discourse relations will further improve the performance of document-level re-annotation.

- The comparison of human and automatic revision was carried out using rough estimates. The annotation efficiency of humans can be impacted by various factors, such as the interface of the annotation platform, the level of skill and training of the annotators as well as their mental condition. The efficiency of automatic re-annotation can also vary depending on the chosen PLMs, the implementation details and the experimental setup. Hence, a more systematic analysis of the difference between human and automatic re-annotators may be conducive to prospective human-central NLP.

Aside from the aforementioned directions, the promising results obtained using our proposed prompt-based credibility score have led us to come up with ideas for extending this work further. Recently, prompt-based methods demonstrate impressive zero-shot learning capability in diverse downstream tasks with impressive zero-shot learning capability (Liu et al., 2021a). Theoretically, we can derive credibility scores on almost all NLP datasets with the appropriate prompts. Under the human-in-the-loop setting, a prompt-based credibility score could help human re-annotators to focus on the most suspicious annotations to speed up their work. Under the noise-tolerant learning setting, the prompt-based credibility score has the potential to improve model performance in various NLP tasks. For instance, when applying curriculum learning (Bengio et al., 2009), the model can greedily learn on the relatively clean data first, and then prudently learn on data that is considered less accurate and consistent by the prompt-based credibility score. We can also design the sampling weight based on this prompt-based approach, encouraging the models to learn on reliable samples most of the time.

Bibliography

Abney, S., Schapire, R. E., and Singer, Y. (1999). Boosting applied to tagging and PP attachment. In Fung, P. and Zhou, J., editors, *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999, College Park, MD, USA, June 21-22, 1999*. Association for Computational Linguistics.

Alex, B., Grover, C., Shen, R., and Kabadjov, M. A. (2010). Agile corpus annotation in practice: An overview of manual and automatic annotation of cvs. In Xue, N. and Poesio, M., editors, *Proceedings of the Fourth Linguistic Annotation Workshop, LAW 2010, Uppsala, Sweden, July 15-16, 2010*, pages 29–37. Association for Computational Linguistics.

Algan, G. and Ulusoy, I. (2021). Metalabelnet: Learning to generate soft-labels from noisy-labels. *CoRR*, abs/2103.10869.

Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127.

Alt, C., Gabryszak, A., and Hennig, L. (2020). TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1558–1569. Association for Computational Linguistics.

Angluin, D. and Laird, P. D. (1987). Learning from noisy examples. *Mach. Learn.*, 2(4):343–370.

Argyris, C. (2004). *Reasons and rationalizations: The limits to organizational knowledge*. Oxford University Press on Demand.

Artstein, R. and Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Comput. Linguistics*, 34(4):555–596.

Aydar, M., Bozal, O., and Özbay, F. (2020). Neural relation extraction: a survey. *CoRR*, abs/2007.04247.

Barnett, V. (1978). The study of outliers: purpose and model. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3):242–250.

Beck, C., Booth, H., El-Assady, M., and Butt, M. (2020). Representation problems in linguistic annotations: Ambiguity, variation, uncertainty, error and bias. In *14th Linguistic Annotation Workshop*, pages 60–73.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Bhadra, S. and Hein, M. (2015). Correction of noisy labels via mutual consistency check. *Neurocomputing*, 160:34–52.

Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *J. Artif. Intell. Res.*, 11:131–167.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bryant, C. J. (2019). *Automatic annotation of error types for grammatical error correction*. PhD thesis, University of Cambridge, UK.

Cao, N. D., Izacard, G., Riedel, S., and Petroni, F. (2021). Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58.

Chaudhuri, D., Rony, M. R. A. H., and Lehmann, J. (2021). Grounding dialogue systems via knowledge graph aware decoding with pre-trained transformers. In Verborgh, R., Hose, K., Paulheim, H., Champin, P., Maleshkova, M., Corcho, Ó., Ristoski, P., and Alam, M., editors, *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, pages 323–339. Springer.

Chen, P., Liao, B., Chen, G., and Zhang, S. (2019). Understanding and utilizing deep neural networks trained with noisy labels. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1062–1070. PMLR.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 1597–1607.

Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., Huang, F., Si, L., and Chen, H. (2021). Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. *CoRR*, abs/2104.07650.

Chi, E. A., Hewitt, J., and Manning, C. D. (2020). Finding universal grammatical relations in multilingual BERT. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5564–5577. Association for Computational Linguistics.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does BERT look at? an analysis of bert's attention. In Linzen, T., Chrupala, G., Belinkov, Y., and

Hupkes, D., editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 276–286. Association for Computational Linguistics.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Crowston, K. (2012). Amazon mechanical turk: A research tool for organizations and information systems scholars. In Bhattacherjee, A. and Fitzgerald, B., editors, *Shaping the Future of ICT Research. Methods and Approaches - IFIP WG 8.2, Working Conference, Tampa, FL, USA, December 13-14, 2012. Proceedings*, volume 389 of *IFIP Advances in Information and Communication Technology*, pages 210–221. Springer.

Cui, L., Wu, Y., Liu, J., Yang, S., and Zhang, Y. (2021). Template-based named entity recognition using BART. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1835–1845. Association for Computational Linguistics.

Cui, M., Li, L., Wang, Z., and You, M. (2017). A survey on relation extraction. In Li, J., Zhou, M., Qi, G., Lao, N., Ruan, T., and Du, J., editors, *Knowledge Graph and Semantic Computing. Language, Knowledge, and Intelligence - Second China Conference, CCKS 2017, Chengdu, China, August 26-29, 2017, Revised Selected Papers*, volume 784 of *Communications in Computer and Information Science*, pages 50–58. Springer.

Davison, J., Feldman, J., and Rush, A. M. (2019). Commonsense knowledge mining from pretrained models. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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, Hong Kong, China, November 3-7, 2019*, pages 1173–1178. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short*

Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dickinson, M. (2010). Detecting errors in automatically-parsed dependency relations. In Hajic, J., Carberry, S., and Clark, S., editors, *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 729–738. The Association for Computer Linguistics.

Dickinson, M. and Lee, C. M. (2008). Detecting errors in semantic annotation. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Dickinson, M. and Meurers, D. (2003). Detecting errors in part-of-speech annotation. In *EACL 2003, 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12-17, 2003, Agro Hotel, Budapest, Hungary*, pages 107–114. The Association for Computer Linguistics.

Dickinson, M. and Smith, A. (2011). Detecting dependency parse errors with minimal resources. In *Proceedings of the 12th International Conference on Parsing Technologies, IWPT 2011, October 5-7, 2011, Dublin City University, Dubin, Ireland*, pages 241–252. The Association for Computational Linguistics.

Dligach, D. and Palmer, M. (2011). Reducing the need for double annotation. In *Proceedings of the Fifth Linguistic Annotation Workshop, LAW 2011, June 23-24, 2011, Portland, Oregon, USA*, pages 65–73. The Association for Computer Linguistics.

Dobbie, S., Strafford, H., Pickrell, W. O., Fonferko-Shadrach, B., Jones, C., Akbari, A., Thompson, S., and Lacey, A. (2021). Markup: A web-based annotation tool powered by active learning. *Frontiers Digit. Health*, 3:598916.

Dozat, T. and Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. *ArXiv*, abs/1611.01734.

Dubey, M. (2021). *Towards Complex Question Answering over Knowledge Graphs*. PhD thesis, University of Bonn, Germany.

Elsayed, G. F., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. (2018). Large margin deep networks for classification. In *Advances in Neural Information Pro-*

cessing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 850–860.

Eskin, E. (2000). Detecting errors within a corpus using anomaly detection. In *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pages 148–153. ACL.

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Trans. Assoc. Comput. Linguistics*, 8:34–48.

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., and Zhou, M. (2020). Codebert: A pre-trained model for programming and natural languages. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1536–1547. Association for Computational Linguistics.

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247.

Fornaciari, T., Uma, A., Paun, S., Plank, B., Hovy, D., and Poesio, M. (2021). Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2591–2597. Association for Computational Linguistics.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Trans. Neural Networks Learn. Syst.*, 25(5):845–869.

Galstyan, A. and Cohen, P. R. (2007). Empirical comparison of ”hard” and ”soft” label propagation for relational classification. In *Inductive Logic Programming, 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers*, pages 98–111.

Gao, M., Zhang, S., Zhang, X., Feng, Z., and Lu, W. (2021a). Graphs and commonsense knowledge improve the dialogue reasoning ability. In Seneviratne, O., Pesquita, C., Sequeda, J., and Etcheverry, L., editors, *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24-28, 2021*, volume 2980 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Gao, T., Fisch, A., and Chen, D. (2021b). Making pre-trained language models better few-shot learners. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Grishman, R. and Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Grivas, A., Alex, B., Grover, C., Tobin, R., and Whiteley, W. (2020). Not a cute stroke: Analysis of rule- and neural network-based information extraction systems for brain radiology reports. In Holderness, E., Jimeno-Yepes, A., Lavelli, A., Minard, A., Pustejovsky, J., and Rinaldi, F., editors, *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis, LOUHI@EMNLP 2020, Online, November 20, 2020*, pages 24–37. Association for Computational Linguistics.

Han, X., Zhao, W., Ding, N., Liu, Z., and Sun, M. (2021). PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Haverinen, K., Ginter, F., Laippala, V., Kohonen, S., Viljanen, T., Nyblom, J., and Salakoski, T. (2011). A dependency-based analysis of treebank annotation errors. In

Gerdes, K., Hajicová, E., and Wanner, L., editors, *Computational Dependency Theory [papers from the International Conference on Dependency Linguistics, Depling 2011, Barcelona, Spain, September 2011]*, volume 258 of *Frontiers in Artificial Intelligence and Applications*, pages 47–61. IOS Press.

Hawkins, D. M. (1980). *Identification of Outliers*. Monographs on Applied Probability and Statistics. Springer.

Hayton, P. M., Schölkopf, B., Tarassenko, L., and Anuzis, P. (2000). Support vector novelty detection applied to jet engine vibration spectra. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 946–952. MIT Press.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. (2020). Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735.

Hénaff, O. J. (2020). Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 4182–4192.

Hess, S., Duivesteijn, W., and Mocanu, D. (2020). Softmax-based classification is k-means clustering: Formal proof, consequences for adversarial attacks, and improvement through centroid based tailoring. *CoRR*, abs/2001.01987.

Hewitt, J. and Manning, C. D. (2019a). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Hewitt, J. and Manning, C. D. (2019b). A structural probe for finding syntax in word representations. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4129–4138. Association for Computational Linguistics.

Hickey, R. J. (1996). Noise modelling and evaluating learning from examples. *Artif. Intell.*, 82(1-2):157–179.

Higuchi, T., Saxena, S., Souden, M., Tran, T. D., Delfarah, M., and Dhir, C. (2021). Dynamic curriculum learning via data parameters for noise robust keyword spotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 6848–6852. IEEE.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Hodge, V. J. and Austin, J. (2004). A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126.

Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognit.*, 40(3):863–874.

Hollenstein, N., Schneider, N., and Webber, B. L. (2016). Inconsistency detection in semantic annotation. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. H. (2013). Learning whom to trust with MACE. In Vanderwende, L., III, H. D., and Kirchhoff, K., editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1120–1130. The Association for Computational Linguistics.

Jamison, E. and Gurevych, I. (2015). Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 291–297. The Association for Computational Linguistics.

Jiang, L., Zhou, Z., Leung, T., Li, L., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2309–2318. PMLR.

Jiang, Z., Anastasopoulos, A., Araki, J., Ding, H., and Neubig, G. (2020a). X-FACTR: multilingual factual knowledge retrieval from pretrained language models. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5943–5959. Association for Computational Linguistics.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020b). How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. (2020). Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Khayrallah, H. and Koehn, P. (2018). On the impact of various types of noise on neural machine translation. In Birch, A., Finch, A. M., Luong, M., Neubig, G., and Oda, Y., editors, *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 74–83. Association for Computational Linguistics.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Khurana, D., Koli, A., Khatter, K., and Singh, S. (2017). Natural language processing: State of the art, current trends and challenges. *arXiv preprint arXiv:1708.05148*.

Klie, J.-C., Bugert, M., Boullosa, B., de Castilho, R. E., and Gurevych, I. (2018). The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics.

Kübler, S., McDonald, R., and Nivre, J. (2009). Dependency parsing. *Synthesis lectures on human language technologies*, 1(1):1–127.

Kusendová, J. (2005). Don mcnicol, *A Primer of Signal Detection Theory.* london: Lawrence. erlbaum associates, publishers 2005. *Glottometrics*, 9:89–90.

Larson, S., Cheung, A., Mahendran, A., Leach, K., and Kummerfeld, J. K. (2020). Inconsistencies in crowdsourced slot-filling annotations: A typology and identification methods. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5035–5046. International Committee on Computational Linguistics.

Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Lee, S., Lee, D. B., and Hwang, S. J. (2021). Contrastive learning with adversarial perturbations for conditional text generation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880.

Li, A., Wang, X., Wang, W., Zhang, A., and Li, B. (2019). A survey of relation extraction of knowledge graphs. In Song, J. and Zhu, X., editors, *Web and Big Data - APWeb-WAIM 2019 International Workshops, KGMA and DSEA, Chengdu, China, August 1-3, 2019, Revised Selected Papers*, volume 11809 of *Lecture Notes in Computer Science*, pages 52–66. Springer.

Li, H. and Liu, Q. (2015). Cheaper and better: Selecting good workers for crowdsourcing. In Gerber, E. and Ipeirotis, P., editors, *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California, USA*, pages 20–21. AAAI Press.

Li, P., Qin, Z., Wang, H., Yang, Q., and Shao, J. (2020a). Exploiting inconsistency problem in multi-label classification via metric learning. In Plant, C., Wang, H., Cuzzocrea, A., Zaniolo, C., and Wu, X., editors, *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 1100–1105. IEEE.

Li, T., Gu, J., Zhu, X., Liu, Q., Ling, Z., Su, Z., and Wei, S. (2020b). Dialbert: A hierarchical pre-trained model for conversation disentanglement. *CoRR*, abs/2004.03760.

Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Li, Z., Cai, J., He, S., and Zhao, H. (2018). Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1073–1094. Association for Computational Linguistics.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Liu, T., Wang, K., Chang, B., and Sui, Z. (2017). A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1790–1795.

Liu, W., Tang, J., Liang, X., and Cai, Q. (2021b). Heterogeneous graph reasoning for knowledge-grounded medical dialogue system. *Neurocomputing*, 442:260–268.

Liu, W., Wen, Y., Yu, Z., and Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 507–516.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ma, Q., Lu, B., Murata, M., Ichikawa, M., and Isahara, H. (2001). On-line error detection of annotated corpus using modular neural networks. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Artificial Neural Networks - ICANN 2001, International Conference Vienna, Austria, August 21-25, 2001 Proceedings*, volume 2130 of *Lecture Notes in Computer Science*, pages 1185–1192. Springer.

Mairal, J. (2013). Optimization with first-order surrogate functions. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA,* *USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 783–791. JMLR.org.

Malossini, A., Blanzieri, E., and Ng, R. T. (2006). Detecting potential labeling errors in microarrays by data perturbation. *Bioinform.*, 22(17):2114–2121.

Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S. (2008). Semantic role labeling: an introduction to the special issue.

Matousek, J. and Tihelka, D. (2017). Anomaly-based annotation error detection in speech-synthesis corpora. *Comput. Speech Lang.*, 46:1–35.

Matsumoto, Y. and Yamashita, T. (2000). Using machine learning methods to improve quality of tagged corpora and learning models. In *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000, 31 May - June 2, 2000, Athens, Greece*. European Language Resources Association.

McLachlan, G. J. (1999). Mahalanobis distance. *Resonance*, 4(6):20–26.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. *CoRR*, abs/2111.01243.

Mucherino, A., Papajorgji, P. J., and Pardalos, P. M. (2009). *k-Nearest Neighbor Classification*, pages 83–106. Springer New York, New York, NY.

Murphy, K. P. (2012). *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press.

Nakagawa, T. and Matsumoto, Y. (2002). Detecting errors in corpora using support vector machines. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

Nan, G., Guo, Z., Sekulic, I., and Lu, W. (2020). Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 1546–1557.

Nghiem, M., Baylis, P., and Ananiadou, S. (2021). Paladin: an annotation tool based on active and proactive learning. In Gkatzia, D. and Seddah, D., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pages 238–243. Association for Computational Linguistics.

Nguyen, Q., Valizadegan, H., and Hauskrecht, M. (2014). Learning classification models with soft-label information. *J. Am. Medical Informatics Assoc.*, 21(3):501–508.

Nicholson, B., Zhang, J., Sheng, V. S., and Wang, Z. (2015). Label noise correction methods. In *2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers, Paris, France, October 19-21, 2015*, pages 1–9. IEEE.

Niu, Z., Shi, S., Sun, J., and He, X. (2011). A survey of outlier detection methodologies and their applications. In Deng, H., Miao, D., Lei, J., and Wang, F. L., editors, *Artificial Intelligence and Computational Intelligence - Third International Conference, AICI 2011, Taiyuan, China, September 24-25, 2011, Proceedings, Part I*, volume 7002 of *Lecture Notes in Computer Science*, pages 380–387. Springer.

Nivre, J. (2005). Dependency grammar and dependency parsing. *MSI report*, 5133(1959):1–32.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021a). Pervasive label errors in test sets destabilize machine learning benchmarks. *CoRR*, abs/2103.14749.

Northcutt, C. G., Jiang, L., and Chuang, I. L. (2021b). Confident learning: Estimating uncertainty in dataset labels. *J. Artif. Intell. Res.*, 70:1373–1411.

Nowak, S. and Rüger, S. M. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In Wang, J. Z., Boujemaa, N., Ramirez, N. O., and Natsev, A., editors, *Proceedings of the 11th ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2010, Philadelphia, Pennsylvania, USA, March 29-31, 2010*, pages 557–566. ACM.

Oppenheimer, D. M., Meyvis, T., and Davidenko, N. (2009). Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of experimental social psychology*, 45(4):867–872.

Palmer, M., Gildea, D., and Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Parde, N. and Nielsen, R. D. (2017). Finding patterns in noisy crowds: Regression-based annotation aggregation for crowdsourced data. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1907–1912. Association for Computational Linguistics.

Passonneau, R. J. and Carpenter, B. (2014). The benefits of a model of annotation. *Trans. Assoc. Comput. Linguistics*, 2:311–326.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Pechenizkiy, M., Tsymbal, A., Puuronen, S., and Pechenizkiy, O. (2006). Class noise and supervised learning in medical domains: The effect of feature extraction. In *19th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2006), 22-23 June 2006, Salt Lake City, Utah, USA*, pages 708–713. IEEE Computer Society.

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

Peng, H., Gao, T., Han, X., Lin, Y., Li, P., Liu, Z., Sun, M., and Zhou, J. (2020). Learning from context or names? an empirical study on neural relation extraction. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3661–3672. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M. E., Neumann, M., IV, R. L. L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. 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, Hong Kong, China, November 3-7, 2019*, pages 43–54.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019a). Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P. S. H., Bakhtin, A., Wu, Y., and Miller, A. H. (2019b). Language models as knowledge bases? In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Portelas, R., Colas, C., Weng, L., Hofmann, K., and Oudeyer, P. (2020). Automatic curriculum learning for deep RL: A short survey. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4819–4825. ijcai.org.

Qian, K., Beirami, A., Lin, Z., De, A., Geramifard, A., Yu, Z., and Sankar, C. (2021). Annotation inconsistency and entity bias in multiwoz. In Li, H., Levow, G., Yu, Z., Gupta, C., Sisman, B., Cai, S., Vandyke, D., Dethlefs, N., Wu, Y., and Li, J. J., editors, *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2021, Singapore and Online, July 29-31, 2021*, pages 326–337. Association for Computational Linguistics.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *CoRR*, abs/2003.08271.

Radev, D. R., Qi, H., Wu, H., and Fan, W. (2002). Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322.

Reif, E., Yuan, A., Wattenberg, M., Viégas, F. B., Coenen, A., Pearce, A., and Kim, B. (2019). Visualizing and measuring the geometry of BERT. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8592–8600.

Reiss, F., Xu, H., Cutler, B., Muthuraman, K., and Eichenberger, Z. (2020). Identifying incorrect labels in the conll-2003 corpus. In Fernández, R. and Linzen, T., editors, *Proceedings of the 24th Conference on Computational Natural Language Learning, CoNLL 2020, Online, November 19-20, 2020*, pages 215–226. Association for Computational Linguistics.

Roit, P., Klein, A., Stepanov, D., Mamou, J., Michael, J., Stanovsky, G., Zettlemoyer, L., and Dagan, I. (2020). Controlled crowdsourcing for high-quality QA-SRL annotation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7008–7013. Association for Computational Linguistics.

Saffari, A., Oliya, A., Sen, P., and Ayoola, T. (2021). End-to-end entity resolution and question answering using differentiable knowledge graphs. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4193–4200. Association for Computational Linguistics.

Saunshi, N., Malladi, S., and Arora, S. (2021). A mathematical exploration of why language models help solve downstream tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (1999). Support vector method for novelty detection. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 582–588. The MIT Press.

Sculley, D. and Cormack, G. V. (2008). Filtering email spam in the presence of noisy user feedback. In *CEAS 2008 - The Fifth Conference on Email and Anti-Spam, 21-22 August 2008, Mountain View, California, USA*.

Sebert, D. M. (1997). Outliers in statistical data. *Journal of Quality Technology*, 29(2):230.

Sellam, T., Das, D., and Parikh, A. P. (2020). BLEURT: learning robust metrics for text generation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.

Sen, P., Oliya, A., and Saffari, A. (2021). Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8805–8812. Association for Computational Linguistics.

Sharou, K. A., Li, Z., and Specia, L. (2021). Towards a better understanding of noise in natural language processing. In Angelova, G., Kunilovskaya, M., Mitkov, R., and Nikolova-Koleva, I., editors, *Proceedings of the International Conference on*

Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3September, 2021, pages 53–62. INCOMA Ltd.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. (2020). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Smyth, P. (1996). Bounds on the mean classification error rate of multiple experts. *Pattern Recognit. Lett.*, 17(12):1253–1257.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 254–263. ACL.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2895–2905. Association for Computational Linguistics.

Sohrab, H. (2003). *Basic Real Analysis*. Birkhäuser Boston.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. (2021). Curriculum learning: A survey. *CoRR*, abs/2101.10382.

Stoica, G., Platanios, E. A., and Póczos, B. (2021). Re-tacred: Addressing shortcomings of the TACRED dataset. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13843–13850. AAAI Press.

Stojnic, V. and Risojevic, V. (2021). Self-supervised learning of remote sensing scene representations using contrastive multiview coding. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 1182–1191.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):44–47.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels. In *3rd International Conference on Learning Representations, ICLR 2015*.

Suzuki, K., Kato, Y., and Matsubara, S. (2017). Correcting syntactic annotation errors based on tree mining. *IEICE Trans. Inf. Syst.*, 100-D(5):1106–1113.

Talmor, A., Elazar, Y., Goldberg, Y., and Berant, J. (2020). olmpics - on what language model pre-training captures. *Trans. Assoc. Comput. Linguistics*, 8:743–758.

Tang, H., Cao, Y., Zhang, Z., Cao, J., Fang, F., Wang, S., and Yin, P. (2020). HIN: hierarchical inference network for document-level relation extraction. In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part I*, pages 197–209.

Taylor, S. E. and Gollwitzer, P. M. (1995). Effects of mindset on positive illusions. *Journal of personality and social psychology*, 69(2):213.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.

Thiel, C. (2008). Classification on soft labels is robust against label noise. In *Knowledge-Based Intelligent Information and Engineering Systems, 12th International Conference, KES 2008, Zagreb, Croatia, September 3-5, 2008, Proceedings, Part I*, pages 65–73.

Tibshirani, R. (1996). Journal of the royal statistical society. series b (methodological).

Tsuzuku, Y., Sato, I., and Sugiyama, M. (2018). Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Vafaeikia, P., Namdar, K., and Khalvati, F. (2020). A brief review of deep multi-task learning and auxiliary task learning. *CoRR*, abs/2007.01126.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

van Rooyen, B., Menon, A. K., and Williamson, R. C. (2015). Learning with symmetric label noise: The importance of being unhinged. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 10–18.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wang, H., Lu, G., Yin, J., and Qin, K. (2021). Relation extraction: A brief survey on deep neural network based methods. In Li, Y. and Nishi, H., editors, *ICSIM 2021: 2021 The 4th International Conference on Software Engineering and Information Management, Yokohama Japan, January 16-18, 2021*, pages 220–228. ACM.

Wang, X., Chen, Y., and Zhu, W. (2020). A comprehensive survey on curriculum learning. *CoRR*, abs/2010.13166.

Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., and Han, J. (2019). Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th Interna-*

tional Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 5153–5162.

Weeber, F., Hamborg, F., Donnay, K., and Gipp, B. (2021). Assisted text annotation using active learning to achieve high quality with little effort. *CoRR*, abs/2112.11914.

Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., Cemgil, A. T., Eslami, S. M. A., and Ronneberger, O. (2020). Contrastive training for improved out-of-distribution detection. *CoRR*, abs/2007.05566.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Wu, Y., Shu, J., Xie, Q., Zhao, Q., and Meng, D. (2021). Learning to purify noisy labels via meta soft label corrector. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10388–10396. AAAI Press.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3733–3742.

Xu, B., Wang, Q., Lyu, Y., Zhu, Y., and Mao, Z. (2021). Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14149–14157.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Yao, L., Mao, C., and Luo, Y. (2019a). KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., and Sun, M. (2019b). DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Ye, D., Lin, Y., Du, J., Liu, Z., Li, P., Sun, M., and Liu, Z. (2020). Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

Ye, H., Zhang, N., Deng, S., Chen, M., Tan, C., Huang, F., and Chen, H. (2021). Contrastive triple extraction with generative transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14257–14265. AAAI Press.

Yuen, M., King, I., and Leung, K. (2011). A survey of crowdsourcing systems. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 766–773. IEEE Computer Society.

Zhang, J., Sheng, V. S., Wu, J., Fu, X., and Wu, X. (2015). Improving label quality in crowdsourcing using noise correction. In Bailey, J., Moffat, A., Aggarwal, C. C., de Rijke, M., Kumar, R., Murdock, V., Sellis, T. K., and Yu, J. X., editors, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1931–1934. ACM.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhang, Y., Qi, P., and Manning, C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zhang, Y. and Yang, Q. (2017). A survey on multi-task learning. *CoRR*, abs/1707.08114.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017a). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Zhang, Y., Zhong, V., Chen, D., Angeli, G., and Manning, C. D. (2017b). Position-aware attention and supervised data improve slot filling. In *Empirical Methods in Natural Language Processing*.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: enhanced language representation with informative entities. In Korhonen, A., Traum, D. R., and Màrquez, L., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.

Zhang, Z. and Sabuncu, M. R. (2018). 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*, pages 8792–8802.

Zhao, M., Zhang, Z., Chow, T. W. S., and Li, B. (2014). A general soft label based linear discriminant analysis for semi-supervised dimensionality reduction. *Neural Networks*, 55:83–97.

Zheng, G., Awadallah, A. H., and Dumais, S. T. (2021). Meta label correction for noisy label learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11053–11061. AAAI Press.

Zhong, Z. and Chen, D. (2020). A frustratingly easy approach for entity and relation extraction. *arXiv: Computation and Language*.

Zhong, Z., Friedman, D., and Chen, D. (2021). Factual probing is [MASK]: learning vs. learning to recall. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5017–5033. Association for Computational Linguistics.

Zhou, T., Wang, S., and Bilmes, J. A. (2021a). Robust curriculum learning: from clean label detection to noisy label self-correction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Zhou, W. and Chen, M. (2021). An improved baseline for sentence-level relation extraction. *CoRR*, abs/2102.01373.

Zhou, W., Huang, K., Ma, T., and Huang, J. (2020). Document-level relation extraction with adaptive thresholding and localized context pooling. *arXiv: Computation and Language*.

Zhou, W., Liu, F., and Chen, M. (2021b). Contrastive out-of-distribution detection for pretrained transformers. In Moens, M., Huang, X., Specia, L., and Yih, S. W., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic,* *7-11 November, 2021*, pages 1100–1111. Association for Computational Linguistics.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zou, X., Zhang, Z., He, Z., and Shi, L. (2021). Unsupervised ensemble learning with noisy label correction. In Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., and Sakai, T., editors, *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2308–2312. ACM.