Understanding, Combating, and Leveraging Imperfect Data in Natural Language Processing

Dawei Zhu

A dissertation submitted towards the degree of
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science of Saarland University

Saarbrücken, 2025

DAY OF COLLOQUIUM

24.10.2025

DEAN OF THE FACULTY

Prof. Dr. Roland Speicher

EXAMINATION COMMITTEE

Chair:	Prof. Dr. Vera Demberg
First Reviewer, Advisor:	Prof. Dr. Dietrich Klakow
Second Reviewer:	Prof. Dr. Benjamin Roth
Third Reviewer:	Prof. Dr. Christian Heumann
Committee Member:	Dr. Volha Petukhova

ABSTRACT

Recent advances in deep neural networks (DNNs) have led to remarkable progress in natural language processing (NLP), largely driven by the increasing scale of both model parameters and training data. However, collecting large-scale data often introduces noise—particularly when relying on automated methods such as weak supervision to reduce annotation costs. This noise can cause DNNs to learn incorrect inductive biases and degrade their generalization ability. Therefore, a deep understanding of data noise and the development of robust learning strategies are essential for the effective deployment of DNNs in real-world NLP applications.

In this thesis, we investigate how data noise affects model generalization and propose methods to address it in practical machine learning scenarios. Our main contributions are as follows:

1. We demonstrate that feature-independent noise has only a minimal impact on Pre-Trained Language Models (PLMs), such as RoBERTa, in classification tasks. At the start of fine-tuning, these models tend to ignore the noise and gradually improve their generalization ability. After reaching the point of best performance, the models begin to memorize noise, which leads to a decline in generalization. We apply an early-stopping mechanism guided by a noisy validation set to stop training before noise memorization occurs, and this yields a model with strong generalization. This simple strategy achieves performance comparable to that of more complex noise-handling methods.

2. In contrast, feature-dependent noise presents a greater challenge. In various token and sequence classification tasks, PLMs quickly overfit to this type of noise, and a noisy validation set is no longer reliable for model selection. We demonstrate the necessity of incorporating a small amount of clean validation data to realign the model. To this end, we propose two methods that leverage clean data to enhance performance in the presence of feature-dependent noise.

3. The emergence of large language models (LLMs) has led to a trend of unifying NLP tasks into generative tasks. We extend our research within this context, focusing on machine translation as a representative task. Our findings show that LLMs have inherent translation capabilities that can be elicited through supervised fine-tuning with a small amount of data. However, despite its small size, the quality of this data plays a crucial role: LLMs are highly sensitive to noise during fine-tuning. For exam-

ple, fine-tuning with 32 high-quality parallel samples results in better generalization than using 1,024 medium-quality parallel samples.

4. Previous studies often regard noisy data as a byproduct of reducing annotation costs through automatic processes like weak supervision. We demonstrate that noisy data can be effectively integrated with gold annotations. In particular, by supplementing gold annotations with lower-quality ones, LLMs can be trained to differentiate between these annotations through preference learning. We show that this approach significantly enhances LLM performance in translation tasks.

ZUSAMMENFASSUNG

Aktuelle Fortschritte bei tiefen neuronalen Netzen (Deep Neural Networks, DNNs) haben zu bemerkenswerten Entwicklungen im Bereich der natürlichen Sprachverarbeitung (Natural Language Processing, NLP) geführt, insbesondere durch die Vergrößerung von Modellgrößen und das Training auf umfangreichen Datensätzen. Allerdings führt die Erhebung großskaliger Daten häufig zu Rauscheffekten- insbesondere dann, wenn automatisierte Methoden wie schwache Supervision (Weak Supervision) zur Reduzierung der Annotationskosten eingesetzt werden. Dieses Rauschen kann dazu führen, dass DNNs fehlerhafte induktive Verzerrungen erlernen und ihre Generalisierungsfähigkeit beeinträchtigt wird. Daher sind ein tiefgehendes Verständnis von Datenrauschen und die Entwicklung robuster Lernstrategien essenziell für den erfolgreichen Einsatz von DNNs in realen NLP-Anwendungen.

In dieser Dissertation untersuchen wir den Einfluss von Datenrauschen auf die Generalisierung von Modellen und entwickeln Methoden zur Bewältigung dieser Herausforderung in praktischen maschinellen Lernszenarien. Unsere Hauptbeiträge sind wie folgt:

1. Wir zeigen, dass merkmalsunabhängiges Rauschen nur einen minimalen Einfluss auf vortrainierte Sprachmodelle (Pretrained Language Models, PLMs) wie RoBERTa bei Klassifikationsaufgaben hat. Zu Beginn der Feinabstimmung ignorieren diese Modelle das Rauschen weitgehend, was zu einer verbesserten Generalisierung führt. Allerdings erreicht diese Verbesserung ein Maximum und nimmt anschließend wieder ab, was auf eine zunehmende Memorierung des Rauschens hinweist. Wir zeigen, dass ein frühzeitiger Stopp der Feinabstimmung, gesteuert durch ein verrauschtes Validierungsset, effektiv verhindern kann, dass das Modell Rauschen memoriert. Diese einfache Strategie erzielt eine vergleichbare Leistung wie wesentlich komplexere Methoden zur Rauschbewältigung.

2. Im Gegensatz dazu stellt merkmalabhängiges Rauschen eine größere Herausforderung dar. Bei Token- und Sequenzklassifizierungsaufgaben neigen PLMs dazu, schnell dieses Rauschen zu overfitten, und ein verrauschter Validierungsdatensatz ist für die Modellselektion unzuverlässig. Wir demonstrieren die Notwendigkeit, eine kleine Menge sauberer Validierungsdaten zu nutzen, um das Modell neu auszurichten. Dazu schlagen wir zwei Methoden vor, die saubere Daten integrieren, um die Leistung trotz merkmalabhängigen Rauschens zu steigern.

3. Mit dem Aufkommen großer Sprachmodelle (Large Language Models, LLMs) werden NLP-Aufgaben zunehmend in generative Aufgaben vereinheitlicht. Wir erweitern unsere Untersuchung in diesem Kontext, indem wir uns auf maschinelle Übersetzung als repräsentative Aufgabe konzentrieren. Unsere Ergebnisse zeigen, dass LLMs über inhärente Übersetzungsfähigkeiten verfügen, die mithilfe eines überwachten Fine-Tunings auf Basis einer geringen Datenmenge aktiviert werden können. Trotz des geringen Umfangs spielt die Qualität dieser Daten jedoch eine entscheidende Rolle: LLMs reagieren äußerst empfindlich auf Rauschen während des Fine-Tunings. Beispielsweise führt das Fine-Tuning mit 32 hochqualitativen parallelen Beispielen zu einer besseren Generalisierung als die Verwendung von 1024 parallelen Beispielen mittlerer Qualität.

4. Während frühere Studien verrauschte Daten meist als Nebenprodukt der Kostensenkung durch automatische Verfahren wie schwache Supervision betrachten, zeigen wir, dass sich solche Daten durchaus effektiv mit Gold-Annotationen kombinieren lassen. Durch die Ergänzung der Gold-Annotationen um solche geringerer Qualität kann in LLMs ein Präferenzlernen angestoßen werden, das sie in die Lage versetzt, zwischen verschiedenen Annotationstypen zu unterscheiden. Wir zeigen, dass dieser Ansatz die Leistung von LLMs insbesondere bei Übersetzungsaufgaben deutlich verbessert.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Dietrich Klakow for providing me with the opportunity to pursue my PhD. This allowed me to immerse myself in academia, broaden my horizons both professionally and personally, and shape an experience that will remain a lasting part of my life. His invaluable guidance, continuous support, and patience throughout this journey were instrumental in shaping my research.

I am truly grateful to Xiaoyu Shen for his exceptional mentorship and unwavering support during my PhD journey. His encouragement was especially meaningful during pivotal moments, such as our long phone call at ACL 2022, which helped me navigate self-doubt. I also greatly appreciated our insightful discussions during my 2023 internship at Amazon Berlin, as well as the memorable moments—working late at the Amazon office, debating research ideas, placing bets on whose subway would arrive first as we headed home, exploring Berlin's diverse cuisine, and the bittersweet farewell when he chose to return to China for a professorship. Our collaboration continued in Ningbo in 2024, where I had the opportunity to work alongside his talented students in an experience that was both intense and rewarding. His mentorship and friendship have meant a great deal to me.

My heartfelt thanks go out to everyone I had the privilege of collaborating with: Aditya Mogadala, David Ifeoluwa Adelani, Michael A. Hedderich (also a senior colleague, friend, and research buddy), Aravind Krishnan, Xiaoyu Shen, Marius Mosbach (also an ideal colleague, friend, and research buddy with whom I shared the office), Miaoran Zhang, Vagrant Gautam, Jesujoba Alabi, Fangzhou Zhai, Ernie Chang, Andreas Stephan, Sony Trenous, Bill Byrne, Eva Hasler, Pinzhen Chen, Yirong Sun, Yanjun Chen, and Junyan Lin. I appreciate the chance to work with and learn from each of you. I am also grateful to all current and former members of LSV, including Claudia Verburg, Nicolas Louis, Dana Ruiter, Lukas Lange, Volha Petukhova, Badr M. Abdullah, Julius Steuer, Paloma Garcia de Herreros, Florian Dietz, Alexander Blatt, Anupama Chingacham, and Zena Al Khalili. We were like a family—always supporting one another—and the working atmosphere at LSV was truly fantastic.

I would also like to acknowledge my dear Chinese friends who accompanied me through my time in Saarbrücken: Fangzhou Zhai, Fei Chen, Wei Shi, Hao Wu, Yue Fan, and Ziwei He. Your companionship brought warmth and joy, transforming ordinary nights into unforgettable memories of laughter, home-cooked meals, and online gaming

battles. These gatherings were more than casual get-togethers; they became a constant source of support and encouragement throughout my doctoral studies.

I am profoundly thankful to my parents. I could not have pursued my PhD without their unwavering love and support.

Finally, and most importantly, I want to express my deepest gratitude to my wife, Linwei Li. She has been by my side throughout this long academic pursuit, from my early days as a first-year bachelor's student struggling with Mathematics I to the sleepless nights of navigating the challenges of my PhD. We have witnessed each other's growth and become better versions of ourselves. Thank you for being my constant source of strength and encouragement through every high and low on this journey!

CONTENTS

ACRONYMS

DNN	Deep Neural Network
LLM	Large Language Model
MT	Machine Translation
NER	Named-Entity Recognition
NLP	Natural Language Processing
PLM	Pre-trained language model
LNL	Learning with Noisy Labels
SFT	Supervised fine-tuning

1 INTRODUCTION

Deep Neural Networks (DNNs) have delivered remarkable results across numerous fields, from computer vision to Natural Language Processing (NLP). A key driver of these successes is the availability of large-scale datasets. However, as datasets have grown in size and complexity, ensuring high-quality annotations has become increasingly difficult. In practice, even carefully curated data often contain labeling errors or inconsistencies, leading to what is commonly referred to as *noisy data*. Despite these imperfections, learning from noisy data is not merely an occasional inconvenience; it has become a central challenge as we push to develop more robust and scalable AI systems.

Noisy labels arise for several reasons. First, DNNs are widely recognized as being data-hungry, yet obtaining high-quality manual annotations is expensive, time-consuming, and labor-intensive, making large-scale curation unsustainable (Frénay and Kabán, 2014; Gilardi, Alizadeh, and Kubli, 2023; Hedderich et al., 2021; Song et al., 2022). In response, a variety of automatic annotation methods have been proposed to alleviate this bottleneck. These methods often rely on (semi-)automatic annotation sources, thereby reducing the amount of human effort required (Gilardi, Alizadeh, and Kubli, 2023; Ratner et al., 2017; Ren et al., 2020; Taori et al., 2023). However, automatically generated annotations are usually less reliable than those provided by human experts, leading to an inevitable introduction of noise into the training data.

Second, achieving error-free annotation is far more challenging than it might appear, even for moderate-sized datasets that receive careful scrutiny. For instance, the widely used CoNLL-03 dataset (Tjong Kim Sang and De Meulder, 2003) for Named-Entity Recognition (NER) was shown in multiple studies to contain approximately 5% annotation errors in both its training and test sets (Reiss et al., 2020; Rücker and Akbik, 2023; Wang et al., 2019d), despite careful and professional curation. A similar issue arises in Machine Translation (MT); Xu et al. (2024b) and Zhu et al. (2024) point to a significant number of imperfect reference translations in the WMT22 (Kocmi et al., 2022) test sets, even though these datasets are often regarded as among the highest-quality benchmarks in the field.

Lastly, dealing with imperfect annotations is a necessary stepping stone on the path to superintelligence (Burns et al., 2023; Ji et al., 2024; Wu et al., 2024d). For many complex tasks, even human experts fall short of providing optimal solutions—consider the game of Go, where determining the *best* move is notoriously difficult (Silver et al., 2016).

Superintelligent models must therefore learn to refine and improve upon these suboptimal inputs, inching closer to truly optimal solutions over time.

In summary, noisy data are ubiquitous in real-world applications and often impossible to avoid. Yet, training neural networks directly on such data can seriously degrade their generalization performance (e.g., Han et al., 2018b; Reed et al., 2015; Zhang et al., 2017). This critical issue frames the Learning with Noisy Labels (LNL) problem: *how to develop models that maintain robust generalization despite training on noisy data, accomplished by employing noise-resistant architectures and training techniques*.

In this thesis, we systematically investigate how noisy training data adversely affects model generalization, with a particular emphasis on NLP applications. Building on these insights, we present effective methods designed to mitigate the harmful impact of noise, enabling DNNs to achieve strong performance even when trained on highly noisy datasets. Furthermore, we explore intriguing scenarios in which noisy data paradoxically enhances learning, shedding new light on the relationship between label quality and model learning.

By addressing these issues, this thesis aims to both deepen our understanding of the fundamental challenges posed by noisy annotations and offer practical strategies for overcoming them. Ultimately, effective learning from noisy data is essential to advancing robust AI capabilities, paving the way for future breakthroughs across diverse domains.

1.1 MAIN CONTRIBUTIONS

We summarize the primary contributions of this thesis as follows:

- **Understanding Noisy Labels**: Label noise can be either feature-dependent or feature-independent. We identify distinct learning patterns in Pre-trained language models (PLMs) based on the type of noise. In Chapter 3, we demonstrate that PLMs, such as RoBERTa (Liu et al., 2019), exhibit strong robustness to feature-independent noise. Specifically, these models achieve high generalization performance early in training, before overfitting the noise in the data. We show that a simple early-stopping strategy effectively captures well-generalized models before noise overfitting occurs. These models perform comparably to those obtained with more sophisticated noise-handling approaches. Importantly, this early-stopping can rely on a noisy validation set, eliminating the need for a cleanly annotated one. In contrast, feature-dependent noise poses a greater challenge. In Chapters 4 and 5, we demonstrate that PLMs can quickly fit incorrect labels generated through weak supervision, often doing so faster than they fit clean labels. While early-stopping remains effective

in this context, it requires at least a small, cleanly annotated validation set.

- **Addressing Feature-Dependent Noise**: In Chapter 4, we introduce a meta-learning-based approach that effectively addresses realistic feature-dependent noise induced by weak supervision, achieving state-of-the-art performance on multiple test sets from the WRENCH (Zhang et al., 2021c) benchmark at the time of our method's proposal. Chapter 5 provides a comprehensive analysis of fine-tuning PLMs on data with feature-dependent noise, proposes more realistic problem settings for weak supervision, and fairly evaluates existing noise-handling methods under these new conditions. Based on our findings, we propose a method that, despite its simplicity, is highly competitive in performance compared to other noise-handling approaches. Notably, our method does not introduce any additional hyperparameters beyond those required for standard fine-tuning, simplifying deployment and saving considerable time in model selection.

- **Extending Noise Analysis to Generation Tasks**: While most research in LNL focuses on classification tasks, we extend the analysis to generation tasks, specifically machine translation. We also transition our foundation models to Large Language Models (LLMs) to incorporate advancements in NLP with LLMs. In Chapter 6, we show that although LLMs primarily perform task alignment rather than acquiring new translation skills during Supervised fine-tuning (SFT), even a small amount of noise can significantly mislead the learning process, resulting in lower performance. This highlights the importance of data quality in SFT. Additionally, we discover that LLMs are more robust to such noise when it occurs in low-resource languages that are underrepresented in their pre-training.

- **Leveraging Noise for Improved Translation**: Scaling up parallel data to fine-tune LLMs for machine translation tasks has been shown to yield diminishing returns in translation performance (Xu et al., 2023, 2024b), possibly due to noise in the reference translations. To address this issue, in Chapter 7 we train LLMs to distinguish quality differences among various translations of the same source sentences, rather than solely fitting them to potentially noisy reference translations. Specifically, we generate multiple translations by sampling LLMs, which may contain different types of mistakes and thus be noisy. We then manually annotate the quality of these translations and incorporate this knowledge into LLMs through preference learning. Our approach consistently improves translation performance across different language directions, overcoming the performance plateau associated with fine-tuning. Notably, while additional annotations

are required for preference learning, only a small amount of data needs to be annotated to outperform standard fine-tuning that uses orders of magnitude more parallel data. Therefore, our method efficiently leverages noisy data without imposing a significant annotation burden.

1.2 OUTLINE

The remainder of this thesis is organized as follows: in Chapter 2, we present the necessary background on the problem of LNL. This includes a formal definition of LNL and a comprehensive overview of various noise-handling methods, systematically organized. Additionally, we introduce PLMs and LLMs, discussing their model architectures and training methods. Chapter 3 focuses on feature-independent noise, examining how model performance develops during training in the presence of such noise. In Chapter 4, we detail our proposed meta-learning-based approach for handling noise. Chapter 5 provides a thorough analysis of noise introduced by weak supervision and proposes a simple yet highly effective method tailored to realistic settings. Subsequent chapters, specifically Chapter 6 and Chapter 7, explore generation tasks involving LLMs. We analyze noise in machine translation tasks and introduce a preference learning-based method that leverages lower-quality data to enhance machine translation performance with LLMs. Supplementary material is provided in Appendices A through E.

1.3 PUBLICATIONS

This thesis integrates findings from the following publications:

[1] **Dawei Zhu**, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, Dietrich Klakow (2022). *Is BERT Robust to Label Noise? A Study on Learning with Noisy Labels in Text Classification*. In Proceedings of the Third Workshop on Insights from Negative Results in NLP @ ACL 2022
https://aclanthology.org/2022.insights-1.8/

[2] **Dawei Zhu**, Xiaoyu Shen, Michael Hedderich, Dietrich Klakow (2023). *Meta Self-Refinement for Robust Learning with Weak Supervision*. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)
https://aclanthology.org/2023.eacl-main.74/

[3] **Dawei Zhu**, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, Dietrich Klakow (2023). *Weaker Than You Think: A Critical Look at Weakly Supervised Learning*. In Proceedings of the 61st Annual Meeting

of the Association for Computational Linguistics (ACL 2023, **Oral Presentation**, **Theme Paper Award**)
https://aclanthology.org/2023.acl-long.796/

[4] **Dawei Zhu**, Sony Trenous, Xiaoyu Shen, Dietrich Klakow, Bill Byrne, Eva Hasler (2024). *A Preference-driven Paradigm for Enhanced Translation with Large Language Models*. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2024, **Oral Presentation**)
https://aclanthology.org/2024.naacl-long.186/

[5] **Dawei Zhu**, Pinzhen Chen, Miaoran Zhang, Barry Haddow, Xiaoyu Shen, Dietrich Klakow (2024). *Fine-Tuning Large Language Models to Translate: Will a Touch of Noisy Data in Misaligned Languages Suffice? (EMNLP 2024)*
https://aclanthology.org/2024.emnlp-main.24/

2

BACKGROUND

This chapter provides a comprehensive overview of key concepts and methodologies that underpin this thesis. It starts with the basic definitions and distinctions between feature-independent and feature-dependent noise generation processes. It then discusses how weak supervision offers faster, lower-cost data annotation through external, often noisy labeling sources, linking these ideas with the broader challenge of noise in training data. To address noisy annotations, the chapter reviews a spectrum of noise-handling approaches—ranging from sample selection and reweighting to label correction, teacher-student frameworks, and specialized loss functions. Additionally, it outlines the architecture and training paradigms of Pre-trained language models (PLMs), which are the models used for learning throughout the thesis.

2.1 LEARNING WITH NOISY LABELS

In academic research, data annotations are often assumed to be accurate. However, in real-world applications, this assumption often does not hold. Obtaining high-quality annotations for real-world tasks typically requires human experts, which is expensive, time-consuming, and difficult to scale. As a result, approaches like crowd-sourcing (Bi et al., 2014; Yan et al., 2010) or weak supervision (see Section 2.2 for details) are frequently used to gather labels more quickly and cheaply. Yet, compared with expert-verified annotations, these methods may introduce a substantial amount of noise.

Modern deep neural networks possess immense capabilities but are susceptible to adopting unwanted inductive biases from lower-quality data, which can lead to poor generalization (Sukhbaatar et al., 2015; Zhang et al., 2017). As demonstrated in (Zhang et al., 2017), deep neural networks can fit arbitrary data distributions with sufficient training, achieving a training loss close to zero. This is particularly problematic when dealing with noisy labels, as the models tend to overfit by memorizing the noise—a phenomenon we term "noise memorization". Therefore, it is crucial to prevent noise memorization in learning with noisy labels.

Formally, let $\mathcal{X}$ represent the feature space and $\mathcal{Y}$ represent the label space. In standard machine learning tasks, a training set of size N, denoted as $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, is sampled from the data generation distribution $D_{X,Y}$ of $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. We refer to the examples in $\mathcal{D}_{\text{train}}$ as *clean examples*, and their labels as *clean labels*.

In the context of Learning with Noisy Labels (LNL), however, the training data is drawn from a noisy distribution $\hat{D}_{X,\hat{Y}}$ of $(X \times \hat{Y}) \in \mathcal{X} \times \mathcal{Y}$, represented as $\hat{\mathcal{D}}_{\text{train}} = \{(x_i, \hat{y}_i)\}_{i=1}^{N}$, and $\hat{\mathcal{D}}_{\text{train}}$ is what the learning algorithm sees. For each training example x_i, its *noisy label* $\hat{y}_i$ may or may not match the ground truth label y_i. The *noise level* of the training set is quantified by $\sum_{i=1}^{N} \mathbb{1}(y_i = \hat{y}_i)$. However, in real-world scenarios, where ground truth annotations are often unavailable, the noise level remains unknown.

Additionally, a validation set, $\mathcal{D}_{\text{val}} = \{(x_m, y_m, \hat{y}_m)\}_{m=1}^{M}$, is typically provided for model selection, while a test set, $\mathcal{D}_{\text{test}} = \{(x_l, y_l, \hat{y}_l)\}_{l=1}^{L}$, is used for evaluation purposes. The goal of learning with noisy labels is to train models that perform well on $\mathcal{D}_{\text{test}}$ (based on the clean labels in $\mathcal{D}_{\text{test}}$), while being trained on $\hat{\mathcal{D}}_{\text{train}}$.

It is generally assumed that clean labels are available in $\mathcal{D}_{\text{val}}$, in Chapter 5, we challenge this assumption's practicality and introduce a more realistic problem setting.

It is worth noting that for generative tasks, such as machine translation, Y_i can represent a sequence of labels or tokens. In such cases, defining the noise level is not straightforward.

2.1.1 *Feature-independent noise*

Feature-independent noise is studied in various works across computer vision and Natural Language Processing (NLP) (Han et al., 2018b; Jindal et al., 2019; Li, Socher, and Hoi, 2020; Merdjanovska, Aynetdinov, and Akbik, 2024; Natarajan et al., 2013; Zhang et al., 2017, i.a.). This line of work is built on the assumption that the noise generation process operates independently of the features, i.e., $P(\hat{Y}|Y, X) = P(\hat{Y}|Y)$. Such an assumption holds in scenarios where class labels share similar semantics, causing annotators to assign these labels in a relatively arbitrary way that does not correlate with the features. A key advantage of this framework is its simplicity: researchers can generate noisy datasets with precisely controlled noise levels by systematically flipping ground truth labels in clean datasets. This capability enables rigorous evaluation of model performance across varying noise ratios.

Two noise types are commonly studied: *symmetric noise* (Jindal, Nokleby, and Chen, 2017; Rooyen, Menon, and Williamson, 2015) and *asymmetric noise* (Jindal, Nokleby, and Chen, 2017; Natarajan et al., 2013; Patrini et al., 2016; Reed et al., 2015). class-conditional Note that symmetric noise is also often referred to as *uniform noise* and asymmetric noise is often called *class-conditional noise*.

A widely accepted assumption with feature independent noise is that$P(\hat{Y} = Y \mid Y) > \max_{\hat{Y} \neq Y} P(\hat{Y} \mid Y)$ (Chen et al., 2020). However, exceptions to this assumption can occur. For instance, when clean training examples are available, they can be used to estimate $P(\hat{Y} = Y \mid$

Y) and use the information to correct the noisy labels, See (Hedderich, Zhu, and Klakow, 2021) for example.

2.1.2 *Feature-dependent noise*

The feature-independence assumption simplifies the noise generation process, making it easier to construct noisy datasets with predefined noise levels and to establish theoretical guarantees for proposed methods. However, this assumption may be oversimplified and fail to accurately represent realistic noise generation processes (Jiang et al., 2020; Merdjanovska, Aynetdinov, and Akbik, 2024; Xiao et al., 2015; Zhang et al., 2021d; Zhu, Liu, and Liu, 2021).

For instance, Hedderich et al. (2020) constructed news text classification datasets using an automated annotation process, where news texts were labeled through keyword matching. Specifically, they created a list of country names, capitals, major cities, and global organizations. Any news text containing terms from this list was automatically assigned the class "World news". This method clearly introduces feature-dependent noise.

Handling feature-dependent noise is more challenging than dealing with feature-independent noise, as evidenced by numerous studies (Hedderich, Zhu, and Klakow, 2021; Merdjanovska, Aynetdinov, and Akbik, 2024; Zhu et al., 2022). Given its prevalence in practical scenarios, this thesis focuses on feature-dependent noise, detailed in Chapters 3 through 7.

2.2 WEAK SUPERVISION

Weak supervision is proposed to reduce the human annotation effort. It involves the (semi-) automatic annotation of unlabeled data using external, noisy supervision sources, commonly referred to as *weak labeling sources* or simply *weak sources*. These sources can include rule-based heuristics, predictions from other (typically smaller) models, or external knowledge bases. Ratner et al. (2017) formalized weak labeling sources as *labeling functions*, which map unlabeled inputs to labels. A closely related concept is distant supervision, which utilizes external resources, such as knowledge bases, to annotate data automatically. For example, documents can be annotated with entities linked to specific categories from external knowledge. Although weak supervision and distant supervision are sometimes used interchangeably, several works (Lison et al., 2020; Mintz et al., 2009; Ratner et al., 2017) distinguish distant supervision as a subset of weak supervision that explicitly relies on structured external resources.

An illustrative application of weak supervision can be found in the work of Lison et al. (2020), where a set of labeling functions was developed for NER annotation. These functions included: a) Small NER

models trained on out-of-domain NER tasks. b) Gazetteers containing lists of country names, person names, and other entities. c) Heuristic functions leveraging features such as casing, part-of-speech tags, dependency relations, and regular expressions. d) Document-level label consistency checks to ensure that named entities appearing multiple times within a document are consistently categorized. Another notable example of weak supervision is in text classification. For instance, Ren et al. (2020) developed eight regular expressions to automatically label the IMDB dataset (Maas et al., 2011) for sentiment analysis. Similarly, they used four regular expressions to annotate the AGNews dataset for a 4-class text classification task. In computer vision, large amounts of noisily annotated data can be obtained by retrieving images from search engines (Chen, Shrivastava, and Gupta, 2013; Fan et al., 2010; Schroff, Criminisi, and Zisserman, 2011).

When annotating data using multiple weak sources, each data point may have zero weak labels (no weak sources can be applied), one weak label (only a single weak source is activated), or multiple weak labels (several weak sources are activated). When multiple weak labels are available, an aggregated weak label can be obtained through majority voting (Yu et al., 2021; Zhu et al., 2023a) or by employing an external label aggregation network (Guan et al., 2018; Ratner et al., 2017; Yan et al., 2016).

Compared to the manual annotations through human experts, annotated labels provided by weak supervision contain more mistakes (i.e., noise in the data), and the they are referred to as *weak labels*. This bridges weak supervision with learning with noisy labels. *Weakly supervised learning* aims to train models that generalize well despite being trained with lower-quality weak labels.

For text classification tasks, two weak supervision benchmarks are commonly used: WRENCH (Zhang et al., 2021c) and WALNUT (Zheng et al., 2022a). More recently, (Merdjanovska, Aynetdinov, and Akbik, 2024) introduced NoiseBench, which focuses on NER tasks and incorporates six types of realistic noise, including weak supervision noise.

2.2.1 *Beyond rule-based weak supervision*

In widely recognized benchmarks such as WRENCH (Zhang et al., 2021c) and WALNUT (Zheng et al., 2022a), annotations are predominantly generated using predefined rules, such as regular expressions. While this approach is straightforward, it significantly limits the applicability of weak supervision in generative NLP tasks. With recent advancements in Large Language Models (LLMs), there is a growing trend of leveraging LLMs as powerful tools for data synthesis across various NLP applications, including RAG (Asai et al., 2023; Tang and Yang, 2024; Zhang et al., 2024b), LLM-based agents (Liu et al., 2024b;

Qin et al., 2023), and general alignment (Dong et al., 2024; Honovich et al., 2022; Taori et al., 2023; Wang et al., 2023). Notably, the quality of annotations generated by LLMs may surpass those produced by humans. For instance, Gilardi, Alizadeh, and Kubli (2023) demonstrates that ChatGPT delivers more accurate labels across various classification tasks. Similarly, Xu et al. (2024b) and Zhu et al. (2024) reveal that a significant portion of reference translations in the widely recognized WMT22 test sets (Kocmi et al., 2022) contain errors, with translations generated by recent LLMs often surpassing these human-produced references in accuracy. Note that, while Chapter 7 explores LLM-generated translations, the primary focus of this thesis does not lie in learning from LLM-generated synthetic data. Nonetheless, this area holds significant potential for future research.

2.3 NOISE-HANDLING METHODS

Neural networks are susceptible to annotation errors; fitting the noisy training data $\hat{\mathcal{D}}_{train}$ can lead to poor generalization (Rolnick et al., 2017; Sukhbaatar et al., 2015; Tanaka et al., 2018; Zhang et al., 2017). To mitigate this issue, various noise-handling methods have been proposed with the objective of training models that generalize well despite being trained on noisy data.

In the following, we introduce a wide spectrum of noise-handling approaches. These methods tackle noise from different perspectives, often leveraging certain empirical observations to identify and address incorrectly labeled examples. First, neural networks tend to fit clean examples more quickly than wrongly labeled ones and exhibit lower losses on clean data (Han et al., 2018b; Yu et al., 2021; Zhu et al., 2022, i.a.). Second, predictions for wrongly labeled examples tend to fluctuate (Chen et al., 2021; Nguyen et al., 2019; Song, Kim, and Lee, 2019, i.a.).

To provide a clearer structure, we categorize noise-handling methods into distinct groups. However, these categories are not mutually exclusive, as more recent approaches often combine multiple strategies to enhance performance. For example, while small-loss examples can be retained for learning, larger-loss examples may be relabeled instead of being directly discarded (Li, Socher, and Hoi, 2020; Mandal, Bharadwaj, and Biswas, 2020; Zhou, Wang, and Bilmes, 2021, i.a.).

2.3.1 *Sample selection*

This area of research aims to mitigate the impact of noisy labels by filtering out incorrectly labeled examples from the training set. These methods commonly leverage the empirical observation that neural networks tend to have smaller losses for correctly labeled examples

(i.e., $\hat{y}_i = y_i$), whereas examples with incorrect labels typically incur higher losses.

Malach and Shalev-Shwartz (2017) introduce an approach where two models are trained simultaneously, updating their parameters only for examples where the two networks disagree on the predicted labels. As training progresses, the area of disagreement between the models shrinks, resulting in fewer updates during later stages and effectively reducing noise memorization, which is more likely to occur in the later stages of training.

Co-teaching (Han et al., 2018b) uses a similar dual-model framework. In each training batch, each model selects a subset of examples with the smallest losses and exchanges these examples with the other model for training (a technique often referred to as the "small-loss trick"). However, a limitation of Co-teaching is that the two networks may converge to a consensus, causing Co-teaching to degrade into self-training, similar to MentorNet (Jiang et al., 2018). To address this issue, and drawing inspiration from Malach and Shalev-Shwartz (2017), Yu et al. (2019) propose retaining only the examples where the two networks disagree. From this disagreement set, the models exchange examples with smaller losses. Building on Co-teaching, Mandal, Bharadwaj, and Biswas (2020) suggest relabeling high-loss examples using model predictions rather than discarding them. This approach enhances data utilization and improves performance. JoCoR (Wei et al., 2020) extends Co-teaching by jointly considering the loss of each example across both networks, mitigating error accumulation caused by biased sample selection within a single network. Co-learning (Tan et al., 2021) employs a shared encoder for both networks, ensuring mutual constraint and maximizing agreement in the latent space, which has been shown to be robust under high noise conditions.

DivideMix (Li, Socher, and Hoi, 2020) also adopts a two-network framework similar to Co-teaching but incorporates semi-supervised learning techniques such as MixMatch (Berthelot et al., 2019). It further refines the process by relabeling examples that are likely mislabeled. Unicon (Karim et al., 2022) addresses an issue in Co-teaching where class imbalance can arise in the selected clean set for each batch. To resolve this, Unicon enforces class balance by selecting an equal number of clean samples per class. Additionally, it integrates contrastive learning to generate noise-resilient feature representations, improving both the precision of clean sample selection and the overall model robustness. Similarly, Sup-CL (Li et al., 2022) leverages contrastive learning for noise-robust sample selection. Similarly, Sup-CL (Li et al., 2022) leverages contrastive learning for robust sample selection. Unlike these mini-batch-based methods, Jo-SRC (Yao et al., 2021) selects clean samples globally, using information from the entire dataset to improve accuracy.

Rather than relying on losses computed from noisy labels, Xia et al. (2015) propose using the reconstruction loss of a trained autoencoder. This method leverages the observation that mislabeled examples tend to exhibit significantly higher reconstruction errors. Northcutt, Wu, and Chuang (2017) suggest pruning training examples where the model's confidence falls below a predefined threshold.

Building on the insight that predictions for noisy samples often fluctuate during training, Nguyen et al. (2019) monitor model predictions over time and filter out samples where the noisy labels conflict with the model's predictions. In a related approach, SELF (Nguyen et al., 2020) maintains a self-ensemble of predictions for all examples throughout the training process, removing instances where the annotated labels disagree with the ensemble predictions.

2.3.2 *Sample Reweighting*

Although sample selection can effectively reduce the influence of noise, it may discard valuable training examples, particularly when noise levels are high. This limitation can be mitigated through label weighting, which serves as a softer alternative to sample selection. For instance, CleanNet (Lee et al., 2018) assigns weights to noisy examples $x, \hat{y}$ based on the cosine similarity between the image embedding of the noisy example (query) and a reference embedding of class $\hat{y}$. These embeddings are computed using a pre-trained image encoder. Similarly, Huang, Zhang, and Zhang (2020) utilize model confidence to determine weights for noisy examples.

L2R (Ren et al., 2018) and MW-Net (Shu et al., 2019) use meta-learning to find weights for the training data that improve performance on clean validation sets. Building on this, Ghosh and Lan (2021) demonstrate that the dependence on clean examples in MW-Net can be eliminated by replacing the cross-entropy meta-loss in MW-Net with a robust mean absolute error loss. Additionally, Wang et al. (2020) use meta-learning to determine example weights, but with a key difference: the training labels are provided by a teacher network rather than noisy training set labels, resulting in a hybrid approach that combines sample reweighting with label correction.

2.3.3 *Label Correction*

A shared limitation of label filtering and weighting techniques is that model updates rely on noisy labels in the training data, which are potentially incorrect. To address this issue, a line of research focuses on correcting labels before updating model parameters.

One of the earliest approaches, introduced by Reed et al. (2015), updates noisy labels using soft labels—a linear combination of the model's predictions and the provided noisy labels. However, the co-

efficients in this linear combination must be predefined and remain fixed throughout training. To improve on this, Ma et al. (2018) propose dynamic coefficients determined through local intrinsic dimensionality (Houle, 2017), which are updated at each training epoch.

Yi and Wu (2019) integrate label correction into the training process in an end-to-end manner. They perform simultaneous model parameter updates and label corrections at each iteration using backpropagation, gradually refining the noisy labels. Similarly, Song, Kim, and Lee (2019) observe that a model's predictions on a training example are likely accurate if it consistently predicts the same label for that example during training. Based on this observation, they replace a training example's label with the model's prediction if (a) the model exhibits high loss with the noisy label and (b) the predictions on that example remain consistent.

Further refinements include methods like that of Chen et al. (2021), which track model predictions during training and then retrain the model using averaged historical predictions. Zheng et al. (2020) propose replacing the noisy label with the model's own prediction when the model exhibits low confidence in the noisy label but high confidence in an alternative label. Meta-learning techniques by Zheng, Awadallah, and Dumais (2021) and Zhu et al. (2023a) also show promise for correcting noisy labels, demonstrating greater effectiveness than sample reweighting approaches such as those presented by Shu et al. (2019).

Another approach involves training a dedicated label-cleaning network. For instance, Li et al. (2017) and Veit et al. (2017) use a small set of clean data, containing both clean and noisy labels, to train this network. The network then corrects noisy labels in the larger training set, improving label quality. Typically, the corrected labels are not used directly but combined with the original noisy labels. This approach accounts for potential errors by the cleaning network. However, noisy labels are often still used throughout training, which may limit the overall model performance.

2.3.4 *Self-training and teacher-student training*

Self-training is a widely used technique in semi-supervised learning (Dehghani et al., 2017; Lee et al., 2013; Yarowsky, 1995). This approach has also been extended to address challenges in learning with noisy labels. In this context, self-training is often used for label filtering, label correction, or both.

Tanaka et al. (2018) propose a method that alternates between updating model parameters and refining labels, thereby iteratively improving the label quality in the training set. SELF (Nguyen et al., 2020) introduces a strategy to progressively filter out mislabeled examples from noisy datasets.

ASTRA (Karamanolakis et al., 2021) uses a teacher network to aggregate weak labels and generate high-quality pseudo-labels for the student network. Other methods (Dehghani et al., 2018; Liang et al., 2020; Yu et al., 2021) train a teacher network on noisy labels and leverage confidence filtering to reduce error propagation when teaching the student network.

In Chapter 4, we introduce a meta-learning-based label correction method. This method replaces noisy labels with predictions from a teacher network, optimized to maximize the student network's validation performance.

2.3.5 *Loss modification and regularization.*

Natarajan et al. (2013) propose an α-weighted 0-1 loss function for binary classification in the presence of feature-independent noise, where α is determined by the noise level. They demonstrate that the α-weighted Bayes optimal classifier under a noisy distribution coincides with the Bayes optimal classifier under the 0-1 loss for a clean distribution. Label smoothing, as discussed by Lukasik et al. (2020) and Zhang et al. (2021b), acts as a loss correction method by assuming uniform noise. However, label smoothing imposes a fixed smoothed distribution, potentially biasing the model. To address this limitation, Lienen and Hüllermeier (2021) introduce label relaxation, a more flexible approach that allows the model to choose from a broader set of distributions instead of relying on a fixed smoothed distribution, thereby enhancing robustness to data noise and improving calibration.

Ghosh, Manwani, and Sastry (2015) and Han, Tsang, and Chen (2016) demonstrate that ramp loss improves robustness against label noise compared to logistic and hinge losses. However, ramp loss is designed for binary classification and is rarely applied in deep learning-based noise-handling methods. In contrast, Ghosh, Kumar, and Sastry (2017) highlight that the mean absolute error loss (MAE) is more robust than the commonly used cross-entropy loss (CCE) for training neural networks on noisy data. Nevertheless, Zhang and Sabuncu (2018) note that MAE can lead to training difficulties and propose generalized CCE, which combines the strengths of both MAE and CCE. Furthermore, Wang et al. (2019c) observe that models trained with CCE often fail to distinguish between "hard" examples and "incorrectly labeled" examples. To address this issue, they propose Symmetric Cross Entropy (SCE) loss, derived from symmetric KL-divergence, and demonstrate its noise robustness both theoretically and empirically. In another approach, Lyu and Tsang (2020) propose curriculum loss, which serves as a tighter upper bound of noise-robust 0-1 loss and demonstrates enhanced resilience to higher noise levels. Additionally, NEEDLE (Jiang et al., 2021) introduces a noise-aware loss

function that dynamically adapts based on the estimated confidence of noisy labels.

Xia et al. (2021) divide model parameters into critical and non-critical groups, observing that non-critical parameters tend to overfit noise. Consequently, they apply weight decay only to non-critical parameters while updating critical ones using gradient descent. Similarly, Han et al. (2020) categorize training examples within a batch into "good" (likely clean) and "bad" (likely noisy) groups. They perform gradient descent on the good group as usual but apply gradient ascent to the bad group. To promote consistency, Iscen et al. (2022) introduce a regularization term that encourages consistent predictions for neighboring data points in the feature space.

Menon et al. (2020) propose composite loss-based gradient clipping, which enhances the robustness of model updates against label noise. Hu, Li, and Yu (2020) demonstrate that regularizing the distance between network parameters and their initialization can also improve noise robustness. Additional studies (Song et al., 2019; Sun et al., 2019; Zhu et al., 2022) find that early-stopping effectively prevents models from memorizing noise, especially when fine-tuning pre-trained language models. Notably, Chen et al. (2020) argue that early-stopping does not require a clean validation set. However, Zhu et al. (2023b) empirically show that this is not effective for feature-dependent noise.

2.3.6 *Noise matrix*

The Noise Matrix is a specific type of loss correction method commonly used in addressing feature-independent label noise. Under the feature-independence assumption, where noise transition probabilities are represented by a matrix T. Each element T_{ij} in this matrix estimates the probability $P(\hat{Y} = j \mid Y = i)$, where $i, j \in \{1, \ldots, C\}$, and C denotes the total number of classes. Once T is estimated, the classification loss l is adjusted to $T^{-1}l$, provided that T is non-singular.

Several approaches have been proposed for estimating the noise matrix. For instance, Goldberger and Ben-Reuven (2017), Patrini et al. (2017), and Wang et al. (2019b) suggest using models pre-trained on noisy training data. Similarly, Chen and Gupta (2015) suggest for training a network solely on relatively high-quality examples to estimate the noise matrix for lower-quality samples. Similiarly, Hedderich, Zhu, and Klakow (2021) and Liu and Tao (2016), leverage clean training examples, assuming such examples are available.

Other methods estimate the noise matrix using only noisy training data. For example, Bekker and Goldberger (2016), Jindal, Nokleby, and Chen (2017), Luo et al. (2017), and Paul et al. (2019) employ techniques like the EM algorithm or regularization directly applied to the matrix. Han et al. (2018a) incorporate structural priors to constrain the noise matrix, noting that certain mislabelings are less likely (e.g.,

annotators are less prone to mislabel an image of a "dog" as a "truck" compared to a similar class). Additionally, Lange, Hedderich, and Klakow (2019) relax the feature-independence assumption by estimating distinct noise matrices for input groups sharing similar features, thereby accommodating feature-dependent noise.

In cases where each example has multiple noisy labels from different annotators, Rodrigues and Pereira (2018) and Tanno et al. (2019) propose modeling separate noise matrices for each annotator.

It is important to note that the noise matrix is designed to map the model's predictions to noisy labels; however, it does not explicitly encourage the base model to predict clean labels. To address this, Jindal, Nokleby, and Pressel (2019) suggest training the base model to corrected labels. These corrected labels are obtained through a linear combination of the model's predictions and the provided noisy labels, in a way similar to the approach in Reed et al. (2015).

2.4 MODEL ARCHITECTURES

In this thesis, unless otherwise stated, we utilize deep neural networks based on the Transformer architecture (Vaswani et al., 2017) for NLP tasks. Below, we introduce the Transformer architecture and discuss common Transformer-based model families: encoder-only, encoder-decoder, and decoder-only models. Finally, we provide an overview of large language models.

2.4.1 *Transformers*

Transformers (Vaswani et al., 2017) have become the foundation for many state-of-the-art models in NLP, computer vision, and speech processing. Unlike recurrent or convolutional architectures, Transformers rely solely on attention mechanisms to capture contextual relationships between tokens.

A Transformer consists of two main components: an *encoder* and a *decoder*, each composed of a stack of layers (or blocks). In the original design of Transformer (Vaswani et al., 2017), both the encoder and decoder have six layers. During forward computation, the input to the first layer is the sum of the input token embeddings and their positional embeddings (discussed in more detail below). The input to each subsequent layer is the output from the preceding layer. Each layer includes the following submodules:

- **Multi-head self-attention**: Captures dependencies by focusing on different input parts simultaneously.
- **Position-wise feed-forward networks**: Applies separate non-linear transformations to each token.

- **Residual connections and layer normalization**: Helps stabilize and accelerate training.

SELF-ATTENTION MECHANISM The core operation in Transformers is *self-attention*. At each layer, self-attention computes a weighted combination of all input vectors for each position in the sequence, thereby capturing contextual dependencies. Formally, given a sequence of d-dimensional input vectors $X \in \mathbb{R}^{n \times d}$, we map them to queries (Q), keys (K), and values (V) using three learned matrices: $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$, respectively:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}_V \in \mathbb{R}^{d \times d_v}$. Here, d_k and d_v denote the dimensions of the queries/keys and values, respectively.

Self-attention first computes attention weights by taking the dot product of queries and keys, scaling by $\sqrt{d_k}$, and applying a softmax to obtain a probability distribution. The output is a linear combination of the value vectors, weighted by the attention weights:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

Instead of a single attention function, Transformers employ *multi-head attention*, allowing the model to attend to information from different representation subspaces. Specifically, given h heads, we initialize h sets of QKV matrices $\mathbf{W}_Q^i$, $\mathbf{W}_K^i$, and $\mathbf{W}_V^i$ for $i = 1, \cdots, h$, where $\mathbf{W}_Q^i, \mathbf{W}_K^i \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}_V^i \in \mathbb{R}^{d \times d_v}$. Each head computes its own attention:

$$\text{head}_i = \text{Attention}(\mathbf{X}\mathbf{W}_Q^i, \mathbf{X}\mathbf{W}_K^i, \mathbf{X}\mathbf{W}_V^i)$$

The outputs of all h heads are then concatenated and projected back to the original input dimension d:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\,\mathbf{W}_O$$

where $\mathbf{W}_O \in \mathbb{R}^{hd_v \times d}$. In the original Transformer design, it is set that $d_k = d_v = d/h = 64$, $d = 512$, and $h = 8$.

Let $X_{\text{out}} \in \mathbb{R}^d$ denote the output of the self-attention mechanism. It is further processed with a residual connection and layer normalization (Ba, Kiros, and Hinton, 2016):

$$X_{\text{out}} = \text{LayerNorm}(X_{\text{out}} + X)$$

POSITION-WISE FEED-FORWARD NETWORK After the self-attention mechanism, each output position is processed by a position-wise feed-forward network (FFN). This feed-forward operation is performed independently for each position:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{W}_1 X_{\text{out}} + \mathbf{b}_1)\,\mathbf{W}_2 + \mathbf{b}_2$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{\text{ff}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{ff}} \times d}$. The dimension d_{ff} is set to 2048 in the original Transformer design.

POSITIONAL ENCODING The self-attention mechanism itself is invariant to the order of inputs. However, natural languages rely heavily on word and sentence order for meaning. To incorporate positional information into Transformers, sinusoidal positional encodings are introduced, defined as:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

where pos is the position index and i is the dimension index. These positional encodings are added to the input token embeddings before being passed to the first encoder layer. Alternatively, many modern Transformer architectures use learnable positional embeddings, which are additional trainable parameters incorporated into the model.

ENCODER VS. DECODER While both the encoder and decoder share similar building blocks (i.e., multi-head self-attention, feed-forward networks, and residual connections), they differ in how these blocks are employed. The encoder processes the entire input sequence to produce contextualized representations of all tokens. In contrast, the decoder operates in an *auto-regressive* fashion, leveraging two types of attention: a *masked* self-attention mechanism to prevent attending to future tokens. This design enables the decoder to generate target sequences token by token, using both previously generated tokens and the encoder-provided context.

It is important to note that while the introduction to Transformers is based on the original architecture proposed by Vaswani et al. (2017), many recent models retain the core ideas and architectural design but incorporate numerous adaptations for improved performance. For example, many contemporary models use learned positional encodings, such as RoPE (Su et al., 2024).

In fact, using either the encoder or decoder module alone can effectively address a wide range of NLP tasks. For instance, pre-trained encoder models perform well in classification tasks, whereas pre-trained decoder models are designed for generative tasks (For more details on pre-training, refer to Section 2.5). Although there has been ongoing debate about whether certain architectures are better suited for specific tasks, no definitive conclusion has been reached as of the time of writing. Nevertheless, pre-trained decoder-only models have become dominant in the NLP community. These models, which now contain billions of parameters and are trained on massive text corpora, demonstrate exceptional performance across NLP applications.

2.4.2 *Encoder-only models*

One of the most well-known encoder-only models is BERT (Devlin et al., 2019), which shows strong performance on various benchmarks in GLUE (Wang et al., 2019a) after pre-training and downstream fine-

tuning. Since then, many encoder-only PLMs have been proposed, such as RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), and DeBERTa (He et al., 2021), which refine BERT's architecture and/or training objectives to achieve better performance. Additionally, variations of BERT-like encoder-only models have been developed for faster training and/or inference. For example, DistilBERT (Sanh et al., 2019) and TinyBERT (Jiao et al., 2020) are significantly smaller models distilled from BERT while maintaining high performance. Another example is ALBERT (Lan et al., 2020), a lite version of BERT that reduces the model size by up to 90% through parameter sharing, while maintaining comparable performance.

2.4.3 *Encoder-decoder models*

Encoder-decoder architectures leverage the bi-directional attention in the encoder to compute powerful language representations of the input, and produce text using the decoder. A prime example of encoder-decoder models is T5 (Raffel et al., 2020), which reframes every NLP task as a text-to-text problem, allowing it to excel across multiple benchmarks. Similarly, BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020) demonstrate strong performance in text generation tasks by leveraging noise-based pre-training objectives that encourage the model to reconstruct corrupted input sequences.

2.4.4 *Decoder-only models*

One prominent family of decoder-only PLMs is the GPT series (Brown et al., 2020; Radford et al., 2018, 2019). These models adopt an autoregressive approach: given a sequence of tokens, they predict the next token based on the previous ones. By training on massive corpora in this left-to-right manner, GPT-like models excel in a variety of generative tasks, including language modeling, text completion, and question answering. Furthermore, they can be adapted for non-generative tasks by framing them as conditional generation problems (e.g., generating a label token for classification).

In addition to the GPT family, a growing number of large decoder-only models exhibit strong capabilities in text generation and beyond. In Chapter 6 and 7, we used decoder-only models including LLaMA (Touvron et al., 2023a), Llama 2 (Touvron et al., 2023b), Mistral (Jiang et al., 2023b), and BLOOM (Muennighoff et al., 2023) for translation tasks.

2.4.5 *Large language models*

Due to the nature of the Transformer design, one can easily expand the model's width by introducing more attention heads or increase

its depth by stacking additional attention blocks. The original Transformer models (Vaswani et al., 2017) had about 65 million parameters (about 213 million for the larger variant), whereas GPT-3 (Brown et al., 2020), with its 175 billion parameters, showcases remarkable zero-shot performance on various NLP tasks. Models with such massive parameter counts, pre-trained on huge data, are often referred to as LLMs (though there is no clear definition of when a model can be considered "large"). Kaplan et al. (2020) present that by increasing the model size, training data, and compute resources, the perplexity can be consistently minimized—a phenomenon known as the scaling law. Wei et al. (2022b) observe that certain capabilities require models of a specific size, referred to as emergent abilities. Refer to (Zhao et al., 2024) for a comprehensive survey of LLMs.

2.5 TRAINING LANGUAGE MODELS

Since 2018, most NLP systems have adopted a two-stage training process. The first stage, known as *pre-training*, involves training a neural language model—often comprising millions to billions of parameters—on large unlabeled corpora. During this phase, the model learns rich token representations that capture linguistic patterns and semantic relationships.

In the second stage, these learned representations are used in supervised training tailored to specific downstream tasks. This approach enables the model to generalize effectively across various applications by leveraging the foundational knowledge acquired during pre-training.

Models developed after the pre-training phase are typically referred to as PLMs. All language models discussed from Section 2.4.2 to Section 2.4.4 fall under this category. A subset of these, known as LLMs, are distinguished by their substantial size and the extensive computational resources—such as large corpora and significant GPU hours—required for their training.

While PLMs can directly address NLP tasks through methods like next-word prediction in an in-context setting (Brown et al., 2020), achieving optimal performance generally necessitates the second stage of supervised training. This subsequent phase has been referred to by various terms over time. Initially termed *fine-tuning* (Devlin et al., 2019), it involves adjusting the pre-trained model using considerably less training data, smaller learning rates, and often only a few training epochs to adapt the model's weights for downstream tasks. Typically, a PLM is fine-tuned on one or a few NLP tasks within the same category (e.g., sentiment analysis).

With the emergence of LLMs, the fine-tuning process has expanded to encompass a wide spectrum of NLP datasets, and this second stage has increasingly been called supervised fine-tuning (Ouyang et al., 2022). In this context, PLMs are fine-tuned on diverse NLP tasks using

supervised methods. Additionally, the development of chatbots, such as (OpenAI, 2023a), has led to framing these NLP tasks in a human instruction format, resulting in the term *instruction tuning*.

Following supervised fine-tuning or instruction tuning, an additional training phase may be applied to better align LLMs with human preferences. This alignment is achieved using algorithms like PPO (Ouyang et al., 2022) and DPO (Rafailov et al., 2023). Recently, both instruction tuning and preference alignment phases have been collectively referred to as *post-training*.

3

HANDLING FEATURE-INDEPENDENT NOISE

In this chapter, we analyze the impact of feature-independent noise on model generalization in natural language understanding (NLU) tasks. We systematically introduce different types and levels of noise into clean training datasets to evaluate how model performance fluctuates across conditions. Our results show that Pre-trained language models (PLMs) exhibit notable robustness to independent noise. Early in the fine-tuning process, generalization improves rapidly, even under severe noise. This improvement will quickly reach a peak and begin to degrade due to noise memorization. This suggests that stopping fine-tuning before memorization begins can yield a well-generalizing model. We also demonstrate that early-stopping, even when using a noisy validation set, can effectively determine this optimal point to stop fine-tuning. Interestingly, contemporary noise-handling techniques primarily slow the rate of decay but contribute little to enhancing peak performance.

The content presented in this chapter is based on:

> **Dawei Zhu**, Michael A Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, Dietrich Klakow (2022). *Is BERT Robust to Label Noise? A Study on Learning with Noisy Labels in Text Classification*. In Proceedings of the Third Workshop on Insights from Negative Results in NLP @ ACL 2022
> URL: https://aclanthology.org/2022.insights-1.8/

3.1 INTRODUCTION

For many languages, domains and tasks, large datasets with high-quality labels are not available. To tackle this issue, cheaper data acquisition methods have been suggested, such as crowd-sourcing or automatic annotation methods like weak and distant supervision. Unfortunately, compared to gold-standard data, these approaches come with more labeling mistakes, which are known as noisy labels. Noise-handling has become an established approach to mitigate the negative impact of learning with noisy labels. A variety of methods have been proposed that model the noise, or clean and filter the noisy instances (Algan and Ulusoy, 2021; Hedderich et al., 2021). Jindal et al. (2019) show, e.g., a 30% boost in performance after applying noise-handling techniques on a CNN-based text classifier.

In a recent work, Tänzer, Ruder, and Rei (2021) showed that BERT (Devlin et al., 2019) has an inherent robustness against noisy labels.

The generalization performance on the clean distribution drops only slowly with the increase of the mislabeled samples. Also, they show that early-stopping is crucial for learning with noisy labels as BERT will eventually memorize all wrong labels when trained long enough. However, their experiments only focus on a single type of noise and a limited range of noise levels. It remains unclear if BERT still performs robustly under a wider range of noise types and with higher fractions of mislabeled samples. Moreover, they perform early-stopping on a clean validation set, which may not be available under low resource settings. Last but not least, they do not compare to any noise-handling methods.

In this chapter, we investigate the behaviors of BERT on tasks with different noise types and noise levels. We also study the effect of noise-handling methods under these settings. Our main results include **(1)** BERT is robust against injected noise, but could be vulnerable to noise from weak supervision. In fact, the latter, even at a low level, can be more challenging than high injected noise. **(2)** Existing noise-handling methods do not improve the peak performance of BERT under any noise settings we investigated; as shown with further analysis, noise-handling methods rarely render the correct labels more distinguishable from the incorrect ones.[1]

3.2 LEARNING WITH NOISY LABELS

PROBLEM SETTINGS. We consider a k-class classification problem. Let D denote the true data generation distribution over $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the feature space and $\mathcal{Y} = \{1, ..., k\}$ is the label space. In a typical classification task, we are provided with a training dataset $S = \{(x_i, y_i)_{i=1}^n\}$ sampled from D. In learning with noisy labels, however, we have no access to D. Instead, a noisy training set $\hat{S} = \{(x_i, \hat{y}_i)_{i=1}^n\}$ sampled from a label-corrupted data distribution $\hat{D}$. The goal is to learn a classifier that generalizes well on the clean distribution by only exploiting $\hat{S}$.

INJECTED LABEL NOISE. To rigorously evaluate noise-handling methods at different noise levels, researchers in this area often construct noisy datasets from clean ones by injecting noise. This can, e.g., reflect annotation scenarios such as crowdsourcing, where some annotators answer randomly or prefer an early entry in a list of options. Modeling such noise is achieved by flipping the labels of the clean instances according to a pre-defined noise level $\varepsilon \in [0, 1)$ and a noise type. There are two commonly used noise types: the single-flip noise (Reed et al., 2015):

1 Our implementation is available on: https://github.com/uds-lsv/BERT-LNL.

$$p_{\text{sflip}}(\hat{y} = j|y = i) = \begin{cases} 1-\varepsilon, & \text{for } i = j \\ \varepsilon, & \text{for one } i \neq j \\ 0, & else \end{cases}$$

and uniform-flip (Rooyen, Menon, and Williamson, 2015) noise

$$p_{\text{uni}}(\hat{y} = j|y = i) = \begin{cases} 1-\varepsilon, & \text{for } i = j \\ \frac{\varepsilon}{k-1}, & \text{for } i \neq j \end{cases}$$

These noise generation processes are feature-independent, i.e., $p(\cdot|y = i, x) = p(\cdot|y = i)$. Therefore, they can be described by a noise transition matrix T with $T_{ij} := p(\hat{y} = j|y = i)$. It is usually assumed that the noise is diagonally-dominant when generating the noisy labels, i.e. $\forall i, T_{ii} > max_{j \neq i} T_{ij}$.

3.3 EARLY-STOPPING ON NOISY VALIDATION SET

When trainied on noisy data without noise-handling, BERT reaches a high generalization performance before it starts fitting the noise. Then it memorizes the noise and the performance on clean distribution drops dramatically (the blue curve in Figure 3.1). Hence, for models without noise-handling, it is crucial to stop training when the generalization performance reaches its maximum.

Tänzer, Ruder, and Rei (2021) use a clean validation set to find this point. However, a clean validation set is often unavailable in realistic low-resource scenarios as it requires manual annotation. Therefore, we use a noisy validation set for early-stopping in all of our experiments and we attain models that generalize well on the clean distribution.

In our example in Figure 3.1, we see that while most noise-handling methods prevent BERT from fitting the noise in the long run, their peak performance is not significantly higher than a vanilla model without noise-handling.

3.4 EXPERIMENTS

DATASET CONSTRUCTION. We experiment with four text classification datasets: two benchmarks, AG-News (Zhang, Zhao, and LeCun, 2015a) and IMDB (Maas et al., 2011), injected with different levels of single-flip or uniform noise; for the weakly supervised noise, we make use of two news topics datasets in two low-resource languages: Hausa and Yorùbá (Hedderich et al., 2020). Hausa and Yorùbá are the second and the third most spoken indigenous language in Africa, with 40 and 35 million native speakers, respectively (Eberhard, Simons, and (eds.), 2019). The noisy labels were gazetteered. For example, to identify

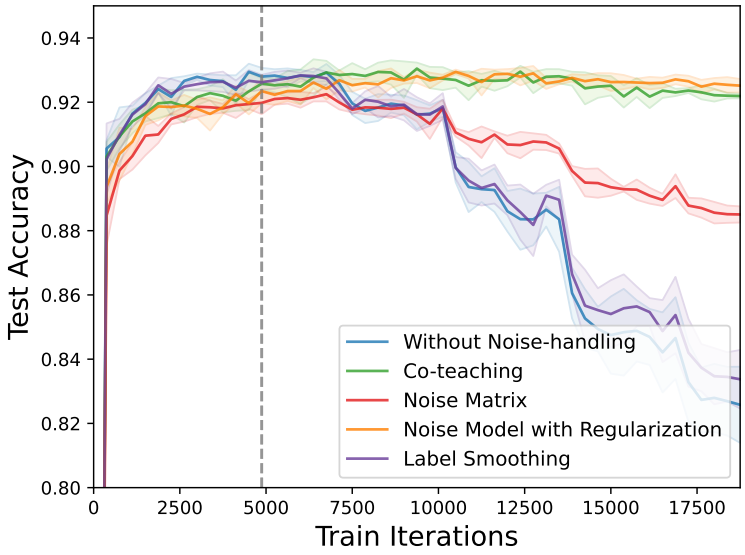


Figure 3.1: A typical training curve when learning with noise. Learning without noise-handling (blue) will reach a peak accuracy before memorizing the noise. Early-stopping on a noisy validation set (vertical grey line) is often sufficient to find such a peak. Injected uniform noise of 40% on AG-News dataset.

texts for the class "Africa", a labeling rule based on a list of African countries and their capitals is used. Note that while we can vary the noise levels of injected noise, the amount of weak supervision noise in Hausa and Yorùbá is fixed[2]. We summarize some basic statistics of the datasets in Table 3.1.

IMPLEMENTATION. We use of-the-shelf BERT models for our tasks. Specifically, we apply the BERT-base model for AG-News and IMDB, and the mBERT-base for Yorùbá and Hausa. The fine-tuning approach follows (Devlin et al., 2019). In all settings, we apply early-stopping on a noisy validation set to mimic the realistic low-resource settings, while the test set remains clean. For more implementation details and a discussion on clean and noisy validation sets, see Appendix A.2 and A.5.

3.4.1 *Baselines*

We compare learning without noise-handling with four popular noise-handling methods.[3]

WITHOUT NOISE-HANDLING Train BERT on the noisy training set as it was clean. A noisy validation set is used for early-stopping.

2 refer to Appendix A.1 for detailed noise distribution.

3 For a fair comparison, early-stopping on a noisy validation set is applied to all four noise-handling methods.

Dataset	Classes	Average Lengths	Train Samples	Validation Samples	Test Samples	Train Noise Level
IMDB	2	292	21246	3754	25000	various
AG-News	4	44	108000	12000	7600	various
Yorùbá	7	13	1340	189	379	33.28%
Hausa	5	10	2045	290	582	50.37%

Table 3.1: Statistics of the text classification datasets. The train noise level is the false discovery rate (i.e., 1-precision) of the noisy labels in the training set. The original AG-News has 120k training instances and no validation instances. We therefore held-out 10% of the training samples for validation.

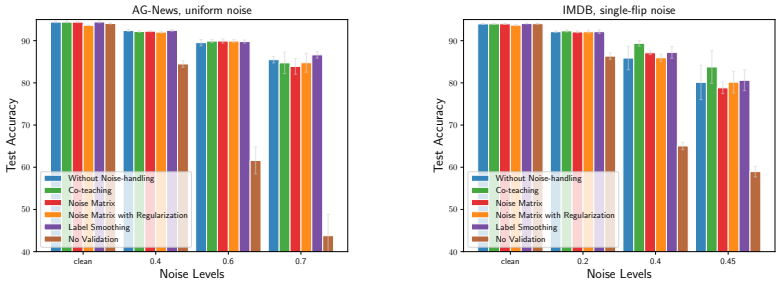
NO VALIDATION For the sake of comparison, we train the model without noise-handling and until the training loss converges.

NOISE MATRIX A noise transition matrix is appended after BERT's prediction to transform the clean label distribution to the noisy one. A variety of methods exists for estimating the noise matrix, i.e. Bekker and Goldberger (2016), Hendrycks et al. (2018), Patrini et al. (2017), Sukhbaatar et al. (2015), and Yao et al. (2020). To exclude the effects of estimation errors in the evaluation, we use the ground truth transition matrix as it is the best possible estimation. This matrix is fixed after initialization.

NOISE MATRIX WITH REGULARIZATION The previous state-of-the-art for text classification with noisy labels (Jindal et al., 2019). Similar to *Noise Matrix*, it appends a noise matrix after BERT's output. During training, the matrix is learned with an $l2$ regularization and is not necessarily normalized to be a probability matrix. In the original implementation they use CNN-based models as backbone, we switch it to BERT for fair comparison.

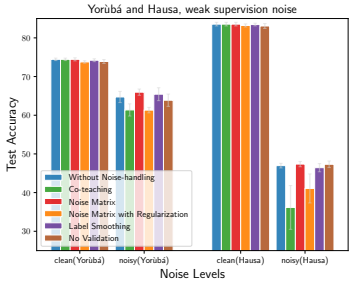
CO-TEACHING Han et al. (2018b) Train two networks to pick cleaner training subsets for each other. The Co-teaching framework requires an estimation of the noise level. Similarly to NMat, we use the ground truth noise level to exclude the performance drop caused by estimation error.

LABEL SMOOTHING Label smoothing (Szegedy et al., 2016) is a commonly used method to improve model's generalization and calibration. It mixes the one-hot label with a uniform vector, preventing the model from getting overconfident on the samples. Lukasik et al. (2020) further shows that it improves noise robustness.



(a) AG-News, uniform noise

(b) IMDB, single-flip noise



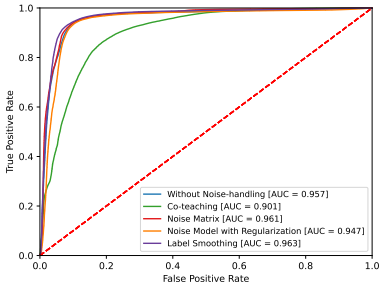
(c) Yorùbá & Hausa, weak supervision noise

Figure 3.2: Test accuracy in different noise settings. a) & b) injected noise with different noise levels c) weak supervision noise, at noise levels of 33.28% and 50.37% in Yorùbá and Hausa, respectively. Noise-handling methods do not always improve peak performances. Further plots in Appendix A.3.

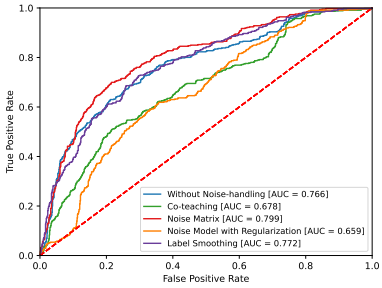
3.4.2 *Experimental Results*

We evaluate our baselines on both injected noise (on AG-News and IMDB) and weak supervision noise (on Hausa and Yorùbá). The test accuracy is presented Figure 3.2. On injected noise, our results match and extend the findings by Tänzer, Ruder, and Rei (2021) that BERT is noise robust. For example, the test accuracy drops only about 10% after injecting 70% wrong labels (Figure 3.2a). However, we find that BERT is vulnerable under weak supervision noise. The performance can drop up to 35% in a dataset like Hausa with 50% weak supervision noise compared to training with clean labels (Figure 3.2c). This indicates that the experience on injected noise may not be transferable to weak supervision noise.

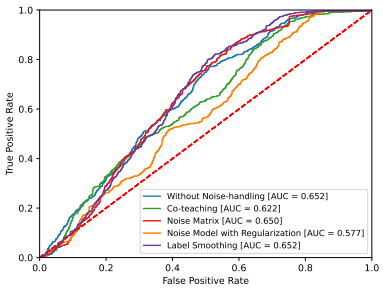
We also observe that noise-handling methods are not always helpful. For injected noise, the benefits from noise-handling become obvious only under high noise levels. But even then, there is no clear winner, meaning that it is hard to decide beforehand which noise method to apply - with the risk that they may even perform worse than BERT without noise-handling. The same applies to weak supervision noise. The maximal performance gap between the best model and BERT



(a) AG-News - 70% uniform noise



(b) Yorùbá - weak supervision noise



(c) Hausa - weak supervision noise

Figure 3.3: ROC curves on wrong label detection (binary classification) using the losses. The losses are recorded at the training step when early-stopping is triggered. Noise-handling methods do not make the losses of correct and incorrect labels more distinguishable. Further plots in Appendix A.4.

without noise-handling is less than 4% and 1.5% under injected noise and weak supervision noise, respectively.

3.4.3 *Analysis of Loss Distributions*

To shed some light on why BERT is robust against injected noise but not weak supervision noise, we track the losses on correctly and wrongly labeled samples during training. Figure 3.4 depicts typical distributions of losses associated with correctly and incorrectly labeled samples, respectively, when early-stopping is triggered. We see that they have minimal overlap, thus different behaviors throughout the training, potentially allowing the model to distinguish correctly and incorrectly labeled samples from each other. We could further quantify the difference by their separability. Figure 3.3 presents the receiver operating characteristic (ROC) curves of a thresholds-based classifier. We observe that **(1)** under injected noise, an area under curve (AUC) of more than 90 can be easily achieved without noise-handling (Figure 3.3a), supporting our observation that injected noise has rather a low impact on BERT. **(2)** Under weak-supervision noise, the AUC score

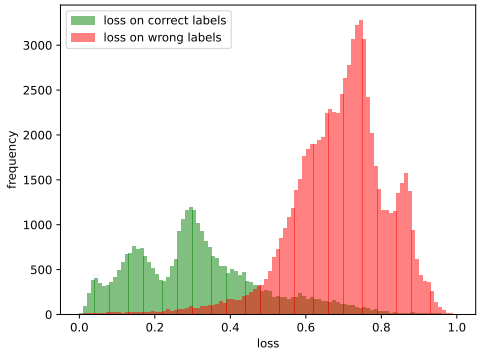


Figure 3.4: Loss histogram at the training iteration when the early-stopping is triggered. AG-News dataset with 70% uniform noise.

is significantly lower, which means the correct and incorrect labels are less distinguishable. Therefore, BERT fits both labels at similar rates. One reason could be that the noise in weak supervision is often feature-dependent, it might become easier for BERT to fit them, which in turn deteriorates the generalization. **(3)** We do not observe a raise in AUC scores when applying noise-handling methods, indicating that noise-handling methods rarely enhance BERT's ability to further avoid the negative impact of wrong labels. This is consistent with the observation in Section 3.4.2 that noise-handling methods have little impact on BERT's generalization performance.

3.5 CONCLUSION

On several text classification datasets and for different noise types, we showed that BERT is noise resistant under injected noise, but not necessarily under weak supervision noise. In both cases, the improvement obtained by applying noise-handling methods are limited. Our analysis on the separability of losses corresponding to correct and incorrect labeled samples provides evidence to this argument. Our analysis offers both motivation and insights to further improve label noise-handling methods and make them useful on more realistic types of noise.

4

A META-LEARNING BASED NOISE-HANDLING METHOD

In the previous chapter, we discussed feature-independent noise, which assumes independence in the noise generation process. This assumption enables researchers to easily construct noisy datasets with different noise levels and perform analyses under controlled settings. However, in many realistic machine learning scenarios, this assumption may oversimplify the noise generation process. For example, weak supervision is an approach that uses various sources to automatically annotate data, saving the time and cost of manual annotation. Since it annotates data based on input features, the resulting noise is feature-dependent. This type of noise is generally more challenging to handle because neural networks can easily detect and reproduce annotation patterns during inference, leading to poor generalization. In this chapter, we present Meta Self-Refinement (MSR), a meta-learning-based noise handling framework that effectively combats feature-dependent noise in various datasets constructed through weak supervision.

The content presented in this chapter is based on:

> **Dawei Zhu**, Xiaoyu Shen, Michael Hedderich, Dietrich Klakow (2023). *Meta Self-Refinement for Robust Learning with Weak Supervision*. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)
> URL: https://aclanthology.org/2023.eacl-main.74/

4.1 INTRODUCTION

Fine-tuning Pre-trained language models (PLMs) has led to great success across Natural Language Processing (NLP) tasks. Nonetheless, it still requires a substantial amount of manual labels to achieve satisfying performance on many tasks. In reality, obtaining large amounts of high-quality labels is costly and labor-intensive (Davis et al., 2013). For certain domains, it is even infeasible due to legal issues and lack of data or domain experts. Weak supervision is a widely-used approach for reudcing such cost by leveraging labels from weak sources, e,g., heuristic rules, knowledge bases or lower-quality inexpensive crowdsourcing (Lison et al., 2020; Ratner et al., 2017; Zhou et al., 2020). It has raised increasing attention in recent years, and efforts have been made to quantify the progress on weakly supervised learning, like the WRENCH benchmark (Zhang et al., 2021c).

Although weak labels are inexpensive to obtain, they are often noisy and inherit biases from weak sources. Training neural networks with weak labels is challenging because of their immense capacity, which leads them to heavily overfit to the noise distribution, resulting in inferior generalization performance (Zhang et al., 2017). Various approaches have been proposed to tackle this challenge. Earlier research focused primarily on simulated noise (Bekker and Goldberger, 2016; Hendrycks et al., 2018), required prior knowledge (Awasthi et al., 2020; Ren et al., 2020) or relied on context-free aggregation rules without leveraging modern pre-trained language models (Fu et al., 2020; Ratner et al., 2017).

Recently, Yu et al. (2021) proposed a contrastive regularized self-training framework that achieved state-of-the-art (SOTA) performance in several NLP tasks from the WRENCH benchmark. It trains a teacher network on weak labels, then iteratively applies the teacher to produce pseudo-labels for training a new student model. To prevent error propagation, it filters the pseudo-labels with the model confidence scores and adds contrastive feature regularization to enforce more distinguishable representations. However, we find that this approach is *effective on easy tasks but fragile on challenging ones*, where the initial teacher model already have memorized a substantial amount of biases with high confidence. Consequently, confidence-based filtering is misleading and all future students will be reinforced with these initial wrong biases from the teacher.

To address this weakness, one strategy is learning to reweight the pseudo-labels with meta learning (Ren et al., 2018; Shu et al., 2019; Wang et al., 2020). By this means, sample weights are dynamically adjusted to minimize the validation loss instead of prefixed with potentially misleading confidence scores. Nevertheless, if the initial teacher is weak and mostly produces incorrect pseudo-labels, simply reweighting the labels does not suffice to extract enough useful training signals.

In this chapter, we propose Meta Self-Refinement (MSR) to go one step further. The teacher is jointly trained with a meta objective such that the student, after one gradient step, can achieve better performance on the validation set. In each training step, a copy of the current student performs one step of gradient descent based on the teacher predictions. The teacher will then update itself towards the gradient direction that minimizes the validation loss of the student. Finally, the actual student is trained by the updated teacher. In MSR, teacher's predictions are iteratively *refined*, instead of only "reweighted", based on the meta objective. This will enable more efficient data utilization since the teacher still has the opportunity to refine itself to provide the proper training signal, even if its initial output label is wrong. To further stabilize the training, we enhance our framework with confi-

dence filtering when teaching the student and apply a linearly scaled learning rate scheduler to the teacher.

In summary, the main contributions are as follows: **1)** We propose a meta-learning based self-refinement framework, MSR, that allows robust learning with label noise induced by weak supervision. **2)** We analyze and quantify how label noise impacts model predictions and representation learning. We find existing methods become less effective in challenging cases when the label noise can be easily fitted. In contrast, MSR is more stable and learns better representation. **3)** Extensive experiments demonstrate that MSR consistently reduces the negative impact of the label noise, matching or outperforming SOTAs on six sequence classification and two sequence labeling tasks.[1]

4.2 RELATED WORK

LEARNING WITH NOISY LABELS. Learning in the presence of label noise is a long-standing problem (Angluin and Laird, 1988). Zhang et al. (2017) show that deep neural networks can memorize arbitrary noise during training, resulting in poor generalization. Noise-handling techniques - by modeling (Goldberger and Ben-Reuven, 2017; Hendrycks et al., 2018; Patrini et al., 2017) or filtering (Han et al., 2018b; Li, Socher, and Hoi, 2020) the noisy instances - are proposed to conquer the label noise. While being effective, they typically assume that the noise is feature-independent which may oversimplify the noise generation process in realistic settings (Gu et al., 2021; Zhu et al., 2022). Recently, realistic and feature-dependent noise induced by weak supervision has received significant attention. To handle this type of noise, Awasthi et al. (2020) propose an implication loss that jointly denoises the noisy labels and weak sources. Ren et al. (2020) denoise the weak label by considering the reliability of different weak sources and aggregating them into one cleaned label. Zhang et al. (2021c) release a benchmark, WRENCH, including various weakly supervised datasets in both text and image domains.

SELF-TRAINING. Self-training (Lee et al., 2013; Yarowsky, 1995) is a simple yet effective framework that is commonly used in semi-supervised learning (SSL). It typically trains a teacher model to provide pseudo-labels for the student model. Different methods have been proposed for better generalization (Mukherjee and Hassan Awadallah, 2020; Xie et al., 2020b; Zoph et al., 2020). Recently, self-training has been adopted to tackle weak supervision. Karamanolakis et al. (2021) train a teacher network that aggregates weak labels to form high-quality pseudo-labels for the student. Liang et al. (2020) and Yu et al. (2021) initialize the teacher model by training a classifier directly on the weak labels, they apply early-stopping to prevent this initial teacher

1 Code is available on: https://github.com/uds-lsv/msr

X = "This film was enjoyable but for the wrong reasons the co-ordination of the action sequences are laughable... and Robert Ginty makes for a film worth seeing."

GT Label
POS

Weak Sources	*HIT?*	*Weak Labels*
Contains(X, laughable)	YES	NEG
Contains(X, enjoyable)	YES	POS
if polarity(X) > 0.8 then pos	YES	POS
RE_Match(X, *highly*recommend *) -> pos	NO	Abstain

Figure 4.1: Sentiment analysis dataset annotated with rule-based weak sources. A weak source is triggered if a specific textual pattern is matched, after which a pre-defined label is then assigned. Otherwise, it abstains. Depending on how many weak sources are triggered, a text may obtain zero, one, or multiple weak labels.

from memorizing the label noise. The student is then trained on the highly confident pseudo-labels provided by the teacher. While the core assumption of self-training - that highly confident pseudo-labels are reliable - is generally valid in SSL, it may not be true for feature-dependent noise induced by weak supervision, especially when the noise is easy to learn. In this case, self-training inevitably suffers more from error propagation and fails to train robust models.

META-LEARNING. Recently, different works leveraged meta-learning techniques to develop noise-robust learning frameworks. The idea is to optimize an outer learner (e.g., sample weights) that guides the inner learner (the classifier) to generalize well. Often, a clean validation dataset is used as a proxy for estimating the generalization performance. Ren et al. (2018) attempt to down weight training samples that increase the validation loss. Shu et al. (2019) employ a neural network to infer such sample weights and show a significant boost on performance under feature-independent noise. Wang et al. (2020) reweight the training samples by their pseudo-labels instead of the original noisy labels. In this chapter, we aim to leverage meta-learning in a more flexible manner by refining the pseudo-labels instead of reweighting them. Approach-wise, the most related works are (Pham et al., 2021; Zhou, Xu, and McAuley, 2022) used for semi-supervised learning and model distillation, which also refine the teacher's parameters based on the student feedback. However, they work with samples from clean distributions, while we anticipate the noise memorization effect and enhance our framework with teacher warm-up and confidence filtering to suppress the error propagation.

4.3 PROBLEM FORMULATION

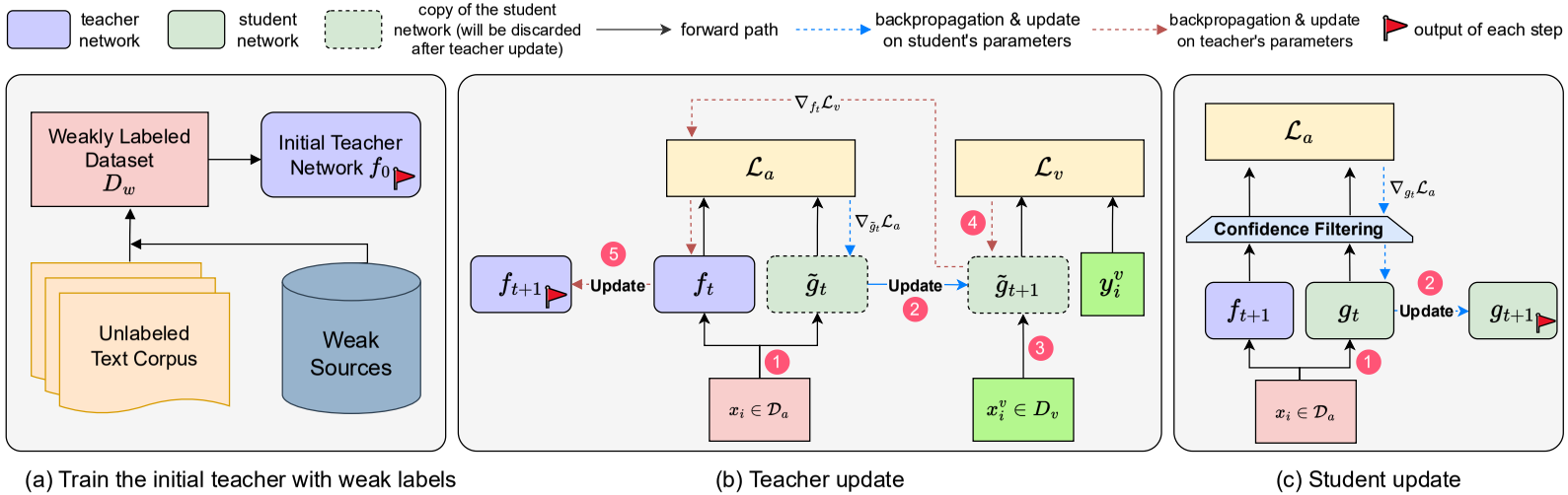
Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label space, respectively. In standard supervised learning, one is given a clean dataset $\mathcal{D}_c = \{(x_i, y_i)\}_{i=1}^N$, where N is the number of samples. The clean labels y_i are supposed to be annotated by human experts.

In weak supervision, a dataset is labeled by weak sources rather than humans. Weak sources can have diverse forms like lexical rules, knowledge bases, pre-trained models, lower-quality inexpensive crowdsourcing, etc. Figure 4.1 shows an example of text labeled via weak supervision. Compared to manual annotations, weak labels contain more mistakes. We denote the dataset labeled by weak sources by $\mathcal{D}_w = \{(x_i, \hat{y}_i)\}_{i=1}^N$ where $\hat{y}_i$ is the weak label.[2] Since weak sources might not cover all data, we may have a set of unlabeled data $\mathcal{D}_u$ in addition to $\mathcal{D}_w$. We use $\mathcal{D}_a = \mathcal{D}_w \cup \mathcal{D}_u$ to denote the full set of data. Moreover, as we do not make any assumption on the quality of the weak labels, their distribution can deviate arbitrarily from the distribution of clean labels. Learning with only weak labels can lead to unbounded model errors (Gu et al., 2021; Menon, Rooyen, and Natarajan, 2016). Hence, following standard practice in weak supervision, we assume the access to a small clean validation set $\mathcal{D}_v = \{(x_i^v, y_i^v)\}_{i=1}^M$ where $M \ll N$. $\mathcal{D}_v$ is used for early-stopping, hyper-parameter tuning or meta-learning so that the learned model will not fully overfit the noisy weak labels (Ren et al., 2018; Shu et al., 2019; Zhang et al., 2021c).

4.4 META SELF-REFINEMENT

We propose a novel meta-learning based framework, named Meta Self-Refinement (MSR), to tackle the label noise induced by weak supervision. In contrast to conventional self-training methods, where the teacher model is fixed after being trained on weakly labeled data, MSR enables the teacher to refine itself based on student performance on the clean validation set, yielding higher-quality labels and more accurate confidence estimates. In this section, we first provide an overview of its training objective (section 4.4.1), then go into the training details (section 4.4.2). Figure 4.2 illustrates the full training process.

2 Multiple weak sources may be triggered simultaneously by a sample. In this case, we can use different aggregation methods like majority voting to determine the final weak label.



(a) Train the initial teacher with weak labels (b) Teacher update (c) Student update

Figure 4.2: Illustration of our proposed Meta-Self Refinement method (MSR). (a) We start by fine-tuning a PLM on weak labels with early-stopping, which yields an initial teacher f_1. (b) At each training step t, f_t gets training signals by performing a "teaching experiment" on $\tilde{g}_t$: a copy of the student network g_t. $\tilde{g}_t$ is updated by fitting f_t with the loss function $\mathcal{L}_a$. f_t is then updated to minimize the validation loss $\mathcal{L}_v$ of $\tilde{g}_{t+1}$. (c): g_t is updated by fitting f_{t+1} with confidence filtering under the loss $\mathcal{L}_a$.

4.4.1 *Training Objective*

MSR contains a teacher network f and a student network g, both are functions that map $\mathcal{X} \rightarrow \mathcal{Y}$. f is initialized by fine-tuning a PLM on the weakly labeled data $\mathcal{D}_w$:

$$f_1 = \arg\min_f \mathbb{E}_{(x_i,\hat{y}_i)\in\mathcal{D}_w} \mathcal{L}(\hat{y}_i, f(x_i)) \tag{4.1}$$

where $\mathcal{L}$ denotes the loss function. We use the cross entropy loss throughout the chapter:

$$\mathcal{L}(p, q) = -\mathbb{E}_{y\sim p(y)} \log q(y) \tag{4.2}$$

p and q are distributions over the label space $\mathcal{Y}$. The initial student network, g_1, is the PLM without fine-tuning on any data.

In conventional self-training, f_1 is used to provide pseudo-labels to train the student. By selecting higher-quality pseudo-labels via confidence filtering (Yu et al., 2021) or uncertainty estimation (Mukherjee and Hassan Awadallah, 2020), the student can often outperform its teacher. However, as the teacher is trained solely on the weak labels, it can easily inherit unexpected biases and provide misleading signals to the student. In MSR, instead of using a fixed teacher to provide pseudo-labels, we use student performance on the clean validation set as a feedback signal to dynamically refine the teacher. Specifically, the objective for the teacher f, formulated as in Equation 4.3, is that *the student network, after fitting the teacher's output labels on $\mathcal{D}_a$, can perform best on the validation set $\mathcal{D}_v$*:

$$\begin{aligned} f^\star &= \arg\min_f \mathbb{E}_{(x_i^v,y_i^v)\in\mathcal{D}_v} \mathcal{L}(y_i^v, g'_f(x_i^v)) \\ g'_f &= \arg\min_g \mathbb{E}_{x_i\in\mathcal{D}_a} \mathcal{L}(f(x_i), g(x_i)) \end{aligned} \tag{4.3}$$

where g' is the student network after fitting output labels from f on $\mathcal{D}_a$. Intuitively, MSR aims to find the best teacher to help the student achieve the lowest validation loss. After finding the optimal teacher $f^\star$ in Equation 4.3, the student can then be obtained by learning from the output labels of $f^\star$:

$$g^\star = \arg\min_g \mathbb{E}_{x_i\in\mathcal{D}_a} \mathcal{L}(f^\star(x_i), g(x_i)) \tag{4.4}$$

4.4.2 *Training Details*

Finding the exact $f^\star$ in Equation 4.3 involves solving two nested loops of optimization, and each loop can be computationally expensive given the large size of $\mathcal{D}_a$. We resort to an online approximation to merge Equation 4.3 and 4.4 into an iterative training pipeline. At each training step t, the teacher f_t is first updated based on the meta-objective of "learning to teach", the student g_t is then trained by the updated teacher.

Algorithm 1: MSR Algorithm

Input: Initial teacher network f_1 trained according to Eq. 4.1. Student network g_1, number of training steps T, teacher's learning rate scheduler $R(t)$, confidence threshold τ, $\mathcal{D}_a$, $\mathcal{D}_v$.

Result: f_T, g_T

for $t \leftarrow 1 \dots T$ **do**

- $\{x_i\} \leftarrow$ SampleMiniBatch($\mathcal{D}_a$)
- $\{x_i^v, y_i^v\} \leftarrow$ SampleMiniBatch($\mathcal{D}_v$)
- `// Teacher Update`
- $\tilde{g}_t \leftarrow \text{Copy}(g_t)$
- $\tilde{g}_{t+1} \leftarrow \tilde{g}_t - \lambda_s \mathbb{E}_{x_i} \nabla_{\tilde{g}_t} \mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$
- $f_{t+1} \leftarrow f_t - R(t) \mathbb{E}_{(x_i^v, y_i^v)} \nabla_{f_t} \mathcal{L}(y_i^v, \tilde{g}_{t+1}(x_i^v))$
- `// Student Update`
- $w(f_{t+1}(x_i)) \leftarrow \mathbb{1}(1 - \frac{H(f_{t+1}(x_i))}{\log(k)} \geq \tau)$
- $g_{t+1} \leftarrow g_t - \lambda_s \mathbb{E}_{x_i} \nabla_{g_t} w(f_{t+1}(x_i)) \mathcal{L}(f_{t+1}(x_i), g_t(x_i))$

end

TEACHER UPDATE. To update the teacher in an efficient way, we approximate the inner loop in Equation 4.3 with a single-step gradient descent of the student network. Namely, the objective of the teacher is changed so that the current student, after *one single gradient descent step* of fitting the teacher, can perform best on the validation set. To do so, the teacher will first conduct a "teaching experiment" on a copy of the current student, denoted as $\tilde{g}_t$. $\tilde{g}_t$ is updated for one gradient descent step to fit the teacher's pseudo labels[3]:

$$\tilde{g}_{t+1} = \tilde{g}_t - \lambda_s \mathbb{E}_{x_i \sim \mathcal{D}_a} \nabla_{\tilde{g}_t} \mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$$

where λ_s is the learning rate of the student network. Afterwards, we update the teacher network to minimize the validation loss of $\tilde{g}_{t+1}$:

$$f_{t+1} = f_t - \lambda_t \mathbb{E}_{(x_i^v, y_i^v) \sim \mathcal{D}_v} \nabla_{f_t} \mathcal{L}(y_i^v, \tilde{g}_{t+1}(x_i^v))$$

where λ_t is the learning rate of the teacher network. It requires calculating second derivatives over f_t. We always use soft labels from the teacher for $\mathcal{L}(f_t(x_i), \tilde{g}_t(x_i))$, so the whole process is fully differentiable. Note that $\tilde{g}_t$ is only used in the "teaching experiment" to help update the teacher. It will be discarded after the teacher is updated.

STUDENT UPDATE. After obtaining f_{t+1}, the real student network is updated with the same objective as in Equation 4.4, except that we use the updated teacher f_{t+1} instead of $f^\star$. As the teacher has performed the "teaching experiment", it will provide more useful signals to guide the student.[4]

3 We use SGD for illustration purposes. The AdamW (Loshchilov and Hutter, 2019) optimizer is used in our experiments.

4 In theory, if the teacher network is strong enough to generalize among different batches, we can directly update the real student in the "teaching experiment", in the

TEACHER LEARNING RATE SCHEDULER. We find the teacher is rather sensitive to its learning rate in practice. If the learning rate is large from the start, the teacher may over-adjust itself due to the large performance gap between the teacher and the student. If the learning rate is small, the teacher will adjust itself too slowly so that more noisy pseudo-labels are passed to the student network. Therefore, we apply a linear learning rate scheduler $R(t) = \frac{t\lambda_t}{T}$ to the teacher network where t denotes the current iteration and λ_t is the targeted learning rate for the teacher. By this means, the teacher's learning rate will gradually increase as it gets better at teaching.

CONFIDENCE-BASED LABEL FILTERING. Despite having the opportunity to refine itself, the teacher inevitably produces some wrong pseudo labels during training, especially at early iterations of self-refinement. To further reduce error propagation, we only select labels with high confidence to guide the student model. The student is updated as follows:

$$g_{t+1} = g_t - \lambda_s \, \mathbb{E}_{x_i \sim \mathcal{D}_a} \nabla_{g_t} \mathcal{L}(f_{t+1}(x_i), g_t(x_i)) \\ \times \mathbb{1}(1 - \frac{H(f_{t+1}(x_i))}{\log(k)} \geq \tau)$$

where $\mathbb{1}$ is the indicator function, $H(f_{t+1}(x_i))$ is the entropy of the distribution $f_{t+1}(x_i)$, k is the number of classes in $\mathcal{Y}$ and τ is a pre-defined confidence threshold. $\log(k)$ is the upper bound of the entropy for k-classification tasks. By this means, only low-entropy (high-confidence) predictions from the teacher are learned. Note that the filtering strategy is only applied to the actual student update step, not during the teaching experiment. Otherwise, the teacher will ignore low-confident samples as they do not contribute to teacher update.

Putting all together, Algorithm 1 summarizes the self-refinement process.

4.5 EXPERIMENTAL SETTINGS

DATASETS. WRENCH (Zhang et al., 2021c) is a well-established benchmark for weak supervision and offers weak labels for various datasets. We compare different baselines on six NLP datasets from WRENCH including both sequence classification and Named-Entity Recognition (NER) tasks. For sequence classification, we include AGNews (Zhang, Zhao, and LeCun, 2015b), IMDB (Maas et al., 2011), Yelp (Zhang, Zhao, and LeCun, 2015b), and TREC (Li and Roth, 2002). For NER tasks, CoNLL-03 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5.0 (Pradhan et al., 2013) are used. In addition, we further include two sequence classification datasets in low-resource languages,

hope that the teacher from the last step can also work in the current batch. However, in practice, we find this mismatch leads to poor performance.

Yorùbá and Hausa (Hedderich et al., 2020), to involve evaluation cases in diverse languages. Table 4.1 summarizes the basic statistics of the datasets. Majority voting over weak sources is used to determine a single label for each sample.

Dataset	Task	# Class	# Train	# Val	# Test
AGNews	Topic	4	96,000	12,000	12,000
IMDB	Sentiment	2	20,000	2,500	2,500
Yelp	Sentiment	2	30,400	3,800	3,800
TREC	Question	6	4,965	500	500
Yoruba	Topic	7	1,340	189	379
Hausa	Topic	5	2,045	290	582
CoNLL03	NER	4	14,041	3,250	3,453
OntoNotes5.0	NER	18	115,812	5,000	22,897

Table 4.1: Dataset statistics. Refer to Appendix B.1 for more details on datasets.

IMPLEMENTATION. RoBERTa-base (Liu et al., 2019) is used as the PLM for English datasets and multilingual BERT-base (Devlin et al., 2019) for non-English ones. We utilize the higher[5] library to perform second-order optimization. Refer to Appendix B.2 for detailed hyper-parameter configurations.

Method	AGNews (Acc)	IMDB (Acc)	Yelp (Acc)	TREC (Acc)	Yorùbá (Acc)	Hausa (Acc)	CoNLL-03 (F1)	OntoNotes (F1)
Fully-Supervised Result								
FT-CL	92.61	93.20	96.91	96.67	77.24	81.57	92.27	85.74
Label Models								
Majority	63.84	71.04	70.21	60.80	58.05	47.93	60.38	58.92
Snorkel (Ratner et al., 2017)	62.67	71.60	68.92	59.60	62.80	47.94	62.88	58.46
DNN Baselines								
FT-WL	$85.73_{\pm 0.43}$	$83.43_{\pm 0.91}$	$87.71_{\pm 1.46}$	$66.80_{\pm 1.44}$	$64.12_{\pm 0.83}$	$46.13_{\pm 0.43}$	$69.20_{\pm 0.33}$	$67.26_{\pm 0.62}$
FT-WLST† (Lee et al., 2013)	$88.61_{\pm 0.71}$	$89.50_{\pm 0.65}$	$95.32_{\pm 0.70}$	$76.00_{\pm 2.21}$	$67.28_{\pm 1.12}$	$49.22_{\pm 1.39}$	$69.87_{\pm 0.36}$	$64.13_{\pm 1.45}$
L2R (Ren et al., 2018)◇	$87.28_{\pm 1.00}$	$82.76_{\pm 1.59}$	$93.34_{\pm 0.91}$	$83.40_{\pm 2.01}$	$70.45_{\pm 0.69}$	$55.67_{\pm 0.88}$	$79.15_{\pm 1.34}$	$70.66_{\pm 0.74}$
Meta-Weight-Net◇ (Shu et al., 2019)	$85.96_{\pm 0.80}$	$86.72_{\pm 0.50}$	$86.97_{\pm 0.74}$	$69.39_{\pm 1.27}$	$70.00_{\pm 2.12}$	$48.63_{\pm 0.96}$	$69.54_{\pm 1.43}$	$69.11_{\pm 1.20}$
Denoise (Ren et al., 2020)	$83.45_{\pm 0.68}$	$76.22_{\pm 0.92}$	$71.56_{\pm 0.56}$	$61.80_{\pm 1.30}$	$66.10_{\pm 1.52}$	$49.31_{\pm 0.93}$	$72.96_{\pm 0.51}$	$67.64_{\pm 1.06}$
UST† (Mukherjee and Hassan Awadallah, 2020)	$87.78_{\pm 0.59}$	$86.74_{\pm 1.18}$	$91.23_{\pm 0.90}$	$77.20_{\pm 2.29}$	$68.12_{\pm 0.71}$	$47.67_{\pm 0.91}$	$69.48_{\pm 1.69}$	$66.98_{\pm 0.99}$
COSINE† (Yu et al., 2021)	$89.34_{\pm 0.76}$	$\mathbf{90.52}_{\pm 1.06}$	$\mathbf{95.48}_{\pm 0.13}$	$82.60_{\pm 1.09}$	$68.87_{\pm 0.82}$	$49.66_{\pm 1.32}$	$70.60_{\pm 0.87}$	$64.59_{\pm 1.08}$
Our Framework								
Teacher-Init (f_1)	$86.37_{\pm 0.00}$	$85.00_{\pm 0.00}$	$89.92_{\pm 0.00}$	$69.00_{\pm 0.00}$	$65.44_{\pm 0.00}$	$46.74_{\pm 0.00}$	$69.73_{\pm 0.00}$	$68.25_{\pm 0.00}$
MSR† ◇	$\mathbf{89.92}_{\pm 0.64}$	$89.16_{\pm 0.91}$	$95.00_{\pm 0.35}$	$\mathbf{94.80}_{\pm 0.29}$	$\mathbf{72.56}_{\pm 0.78}$	$\mathbf{59.11}_{\pm 0.78}$	$\mathbf{88.41}_{\pm 0.63}$	$\mathbf{74.59}_{\pm 0.84}$

Table 4.2: Accuracy and F1 score (in %) on eight NLP tasks. The mean and standard deviation over five trials are reported. Teacher-Init is the best model checkpoint selected from the five trials of FT-WL (according to the validation performance). For a fair comparison, all self-training-based models use the same Teacher-Init checkpoint. MSR matches or outperforms SOTAs on all tasks. † self-training based method. ◇ meta-learning based method.

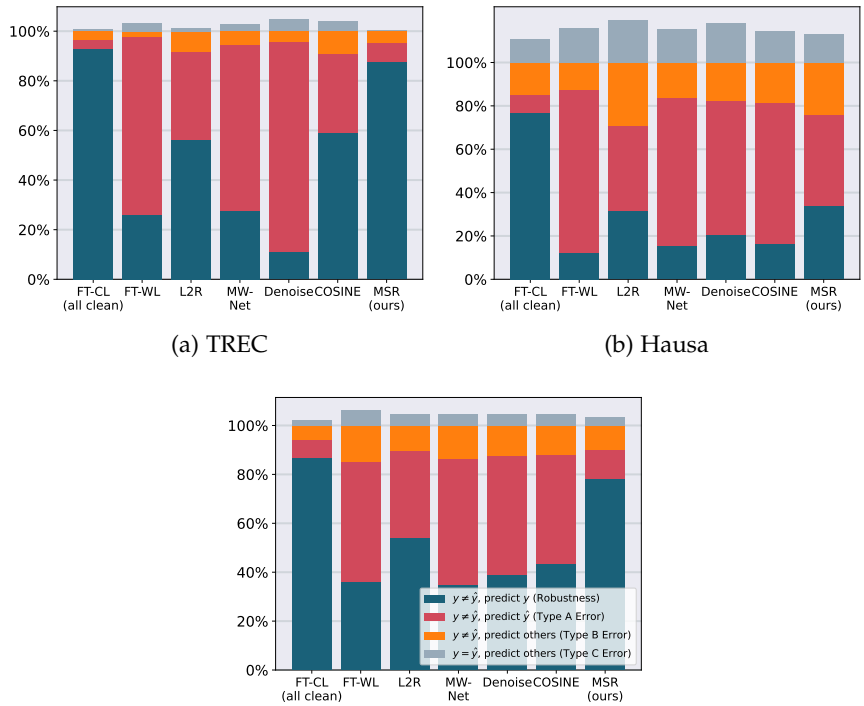
5 https://github.com/facebookresearch/higher

BASELINES. We compare our method with prior work on learning with noisy labels. 1) **Majority** applies majority vote on the weak labels. Ties are broken by randomly selecting a weak label. 2) **Snorkel** (Ratner et al., 2017) trains a labeling model that aggregates weak labels from different weak sources. 3) **FT-WL** fine-tunes PLMs on the weak labels. 4) **FT-WLST** further applies classic self-training (Lee et al., 2013) on the model obtained by FT-WL. 5) **L2R** (Ren et al., 2018) uses a meta-learning framework to reweight weakly labeled samples. 6) **Meta-Weight-Net** (Shu et al., 2019) also applies meta-learning based sample reweighting. However, the weights are computed through an external reweighting network. 7) **Denoise** (Ren et al., 2020) iteratively corrects wrong annotations in the training set, and the classifier learns with the corrected labels. 8) **UST** (Mukherjee and Hassan Awadallah, 2020) is a self-training based method that assigns higher weights to samples that the teacher is certain about. The uncertainties are measured via MC-dropout on the predictions (Gal and Ghahramani, 2016). 9) **COSINE** (Yu et al., 2021) trains its student network with pseudo-labels which the teacher is highly confident about. In addition, contrastive regularization is introduced to further alleviate error propagation.

For our proposed framework, we report the performances of both **Teacher-Init (f_1)**: the initial teacher trained directly on weak labels, and **MSR**: the final student model (g_T). f_1 is obtained by running FT-WL five times and selecting the best one among them according to the validation performance. *For a fair comparison, the same f_1 is used as the initial teacher for all self-training based models*. Finally, we also include the results of fine-tuning PLMs on the clean versions of each dataset, denoted by **FT-CL**, to represent the upper bound performance.

4.6 RESULTS

COMPARISON WITH BASELINES. Table 4.2 shows a comparison among different methods. MSR matches or outperforms SOTAs on all eight datasets. FT-WL outperforms majority voting over the weak labels in all cases except Hausa, which leads to a minor drop. This confirms that PLMs encode useful knowledge in their parameters, enabling them to generalize better than weak rules they are trained on. This phenomenon is particularly noticeable on AGNews, IMDB, and Yelp: direct fine-tuning on the noisy labels (FT-WL) can already achieve decent performance (accuracy above 83%). *We consider them easy tasks since label noise has only a minor impact on performance of PLMs and decent generalization can be attained even without specific noise-handling.* Applying self-training to such simple tasks lead to further performance improvement. COSINE, a SOTA self-training based model, can even perform comparably to the fully supervised model on these three datasets. On the other five datasets, however, FT-WL performs poorly and conventional self-training methods provide little performance



(a) TREC

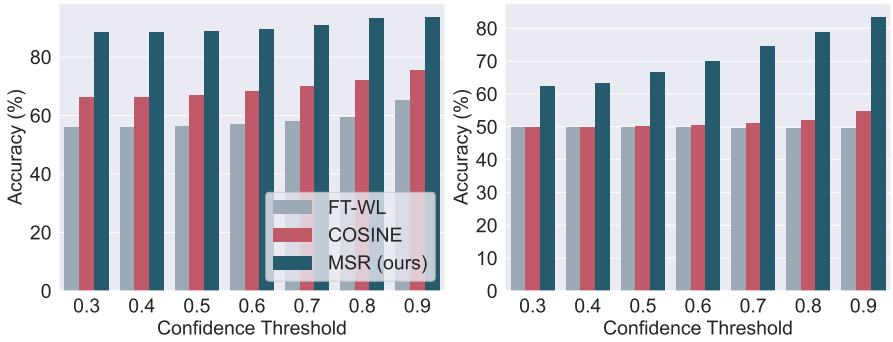
(b) Hausa

(c) CoNLL-03

Figure 4.3: Prediction error decomposition of various weak supervision baselines, evaluated on the test sets. A model is considered robust against label noise if it manages to predict the correct labels despite the wrong weak labels (the robustness is represented by the blue bars). Otherwise, it conforms to the weak label (Type-A error) or predict another incorrect label (Type-B error), which has a negative effect on generalization. The Type-C error rate quantifies the proportion of incorrect model predictions when weak labels are correct. MSR consistently reduces the Type-A error rate and attains a high level of noise robustness.

boost (even a disservice on OntoNotes). This implies that *self-training relies on a well-performed initial teacher to work effectively*. On challenging datasets where the initial teacher is weak, it struggles to achieve further performance gain. Meta-learning based methods such as L2R performs better than COSINE on these challenging datasets. *MSR can further boost the performance on all the challenging datasets by up to 11.4% in accuracy or 9.26% in F1 score while maintaining comparable results on simpler datasets*.

ERROR DECOMPOSITION. Let $y', \hat{y}, y$ denote the model prediction, the noisy weak label, and the clean label, respectively. To investigate how the label noise influences the model predictions, we decompose model prediction errors into three types: (1) Type-A error: $y' = \hat{y}; \hat{y} \neq y$ (2) Type-B error: $y' \neq \hat{y} \neq y$ and (3) Type-C error: $y' \neq y; y = \hat{y}$. Type-A/B errors correspond to situations in which a model complies with



(a) TREC (b) Hausa

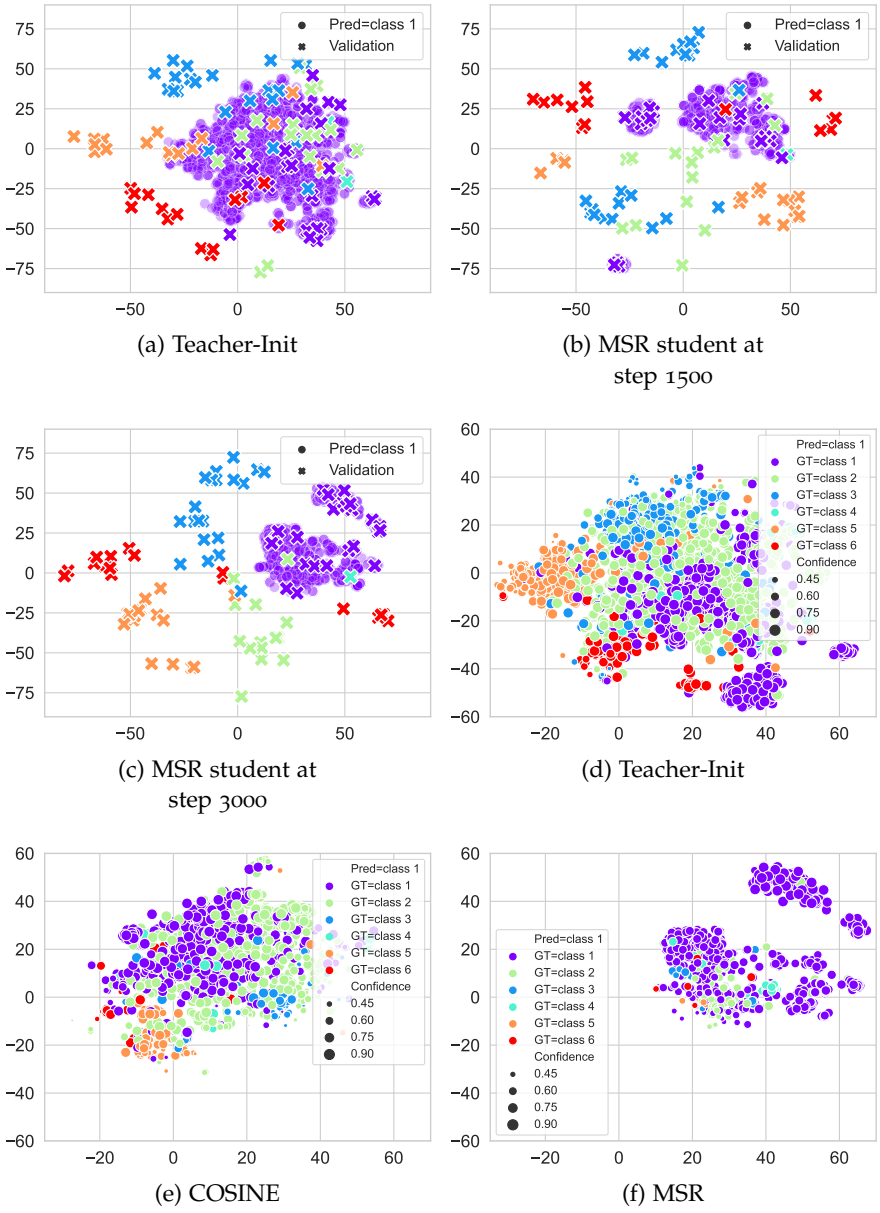
Figure 4.4: Accuracy *vs.* confidence thresholds.

an incorrect weak label $\hat{y}$, or predicts another incorrect class label. If, on the other hand, the weak label $\hat{y}$ is correct, a Type-C error arises if the model predicts a label different than $\hat{y}$. A higher Type-A error rate indicates that a model memorizes more label noise from the weak sources, while a model that underfits fails to learn useful knowledge from the weak sources can have a higher Type-C error rate.

Figure 4.3 visualizes the three types of errors on three challenging datasets: TREC, Hausa and CoNLL-03. The blue bars represent model robustness, i.e., how often the model predicts correctly when $\hat{y} \neq y$. It clearly shows that direct fine-tuning on weak labels (FT-WL) has a much higher Type-A error rate compared with the model trained on clean data (FT-CL), suggesting that the model quickly memorizes the label noise. On the other hand, the disparity in type C error rate is much smaller, indicating that all models do not underfit and the knowledge from the weak sources is properly transferred. The Type-B error shows similar trends and does not differ much across models. Overall, Type-A error has the strongest impact on model performance. *All the noisy-handling models mainly help with reducing Type-A errors*. We also observe that while COSINE reduces Type-A errors on TREC, it barely works on the other two datasets. Only MSR manages to consistently reduce Type-A errors by over 20% on all three datasets.

ACCURACY VS CONFIDENCE. As confidence-based filtering is a key component in both COSINE and MSR, we show the accuracy of model predictions with different confidence thresholds in Figure 4.4. As can be seen, *even using a high confidence threshold for COSINE, the accuracy is still low*, which is why it struggles to improve on challenging datasets. MSR, on the contrary, consistently attains higher accuracy with higher confidence thresholds, and thereby confidence-based filtering on top of MSR help lead to better performance.

IMPACT OF LABEL NOISE ON FEATURE SPACE. We also analyze how the label noise influences representation learning. Figure 4.5



(a) Teacher-Init

(b) MSR student at step 1500

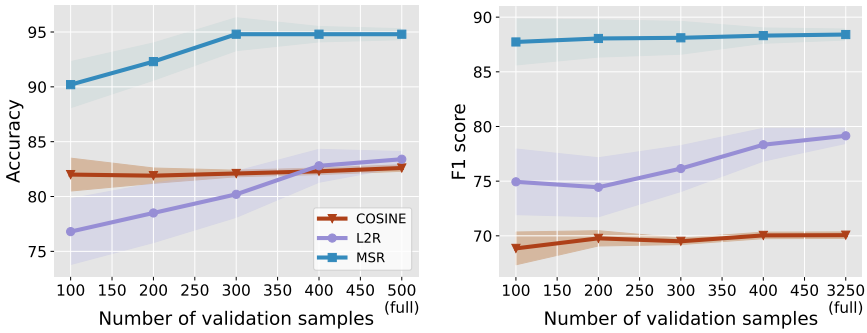
(c) MSR student at step 3000

(d) Teacher-Init

(e) COSINE

(f) MSR

Figure 4.5: Projected feature space of different models on TREC using t-SNE (Maaten and Hinton, 2008). The circles represent training samples that are predicted as class 1. **a)-c)**: development of MSR during training. Circles are colored by the predicted class (i.e., class 1, in purple). The validation samples are represented by crosses and colored according to the ground truth labels. The MSR student gradually improves its feature space to embed the training and validation samples from the same class in the same area. **d)-f)**: training samples are colored according to their ground truth labels; model confidence is reflected by the size of the circles. Teacher-Init and COSINE misclassify samples with high confidence. MSR attains a cleaner cluster.



(a) TREC (b) CoNLL-03

Figure 4.6: Accuracy *vs.* number of validation samples.

illustrates the projected feature space of different models on TREC. For a clear visualization, we present only training samples predicted as class 1 by the models in the form of circles. In figs. 4.5a to 4.5c, we further visualize the feature space of validation samples (represented by crosses). As can be seen, initially the feature space of class 1 overlaps with that of other classes from the validation set. As the training proceeds, when the teacher keeps refining itself, the MSR student gradually reduces such overlap and learns a well-split representation space. In figs. 4.5d to 4.5f, we compare the feature space between different models. The training samples are colored according to their ground truth classes to highlight the misclassification ratio (the more colorful the clusters, the higher the misclassification ratio). We observe that Teacher-Init makes many wrong predictions with high degree of confidence. In this case, utilizing the confidence score for denoising is fragile. This may explain why COSINE, despite offering a more compact cluster, still has a considerable amount of misclassification. Finally, MSR has a considerably cleaner cluster and is less affected by error propagation than COSINE.

EFFECTS OF VALIDATION DATA SIZE. The model performance reported in Table 4.2 is based on the original data splits from the WRENCH benchmark. The size of the validation sets is mostly less than 15% of the training sets. Typically, they are used to perform early-stopping and model selection. For meta-learning based methods, they additionally rely on the validation sets for meta-update and might be more sensitive to validation size. Hence, we study how the validation size affects different models. In particular, we randomly sample a subset from the original validation set $\mathcal{D}_v$ and repeat the same training process. Figure 4.6 presents the results. We find that the validation size indeed has a greater impact on meta-learning approaches. However, *MSR still retains its high generalization performance even with as few as 100 validation samples*, suggesting that MSR is very data efficient in performing the self-refinement.

Configuration	Seq. Classification (Acc)	NER (F1)
Teacher-Init	73.75	68.99
Student	**83.43**	**81.50**
Teacher	82.38 (↓ 1.05%)	80.26 (↓ 1.24%)
w/o Teacher Scheduler	81.80 (↓ 1.63%)	80.15 (↓ 1.35%)
w/o Confidence Filtering	82.32 (↓ 1.11%)	81.09 (↓ 0.41%)
w/o Both	81.63 (↓ 1.80%)	79.95 (↓ 1.55%)

Table 4.3: Summary of ablation experiments aggregated across multiple datasets. See Appendix B.4 for results in each dataset.

ABLATION STUDY. Table 4.3 summarizes the impact of different components of our method. In general, our student model performs slightly better than the teacher. This is as expected because a) the teacher's goal is to guide the student to generalize better, the training loss does not explicitly encourage the teacher to improve its accuracy, and b) the confidence filtering helps the student avoid fitting some wrong pseudo-labels from the teacher. This is also confirmed by the decreased performance when the filter is removed. In addition, applying a learning rate scheduler is better than using a fixed learning rate throughout training.

4.7 CONCLUSION

We present MSR, a meta-learning based self-refinement framework that enables robust learning with weak labels. Unlike conventional self-training which relies on a fixed teacher, MSR dynamically refines the teacher based on the student's performance on the validation set. To further suppress error propagation, we introduce a learning rate scheduler to the teacher and add confidence filtering to the student. We demonstrate that our framework performs on par with or better than current SOTAs on both sequence classification and labeling tasks.

4.8 LIMITATIONS

In this chapter, Our primary focus is to propose a strong weak supervision method that works reliably under various weak supervision settings. We employ meta-learning techniques to address the issue of unreliable confidence scores under challenging settings (Figure 4.4). Despite the effectiveness, the main limitation of our method, just like other meta-learning based frameworks, is the computational overhead. The teacher update step (Algorithm 1, Line 4-6) requires

computing both the first and second-order derivatives, which incurs additional computation time and higher memory consumption. Consequently, our method requires longer training.[6] Implementation-wise and computation-wise, MSR is as complex as other existing meta-learning based methods, like L2R (Ren et al., 2018) and MW-Net (Shu et al., 2019), but performs substantially better than them in all weak supervision scenarios we evaluated. It is worth noting that MSR has *no overhead at inference time*. In weak supervision, the data annotation cost is considered the most significant bottleneck. A stronger model is often obtained by trading some more computation with the cost and effort of obtaining more human-generated, manual annotations. Hence, the one-off investment of training MSR can be worthwhile for real-world weak supervision applications.

6 Detailed training time on each dataset can be found in Appendix B.5 The most costly training of MSR takes roughly 3 hours.

5

REALISTIC FEATURE-DEPENDENT NOISE HANDLING

In few-shot learning scenarios for NLP tasks, recent studies have demonstrated that using a validation set with significantly more examples than the few training shots undermines the feasibility of few-shot learning and violates its core assumptions (Perez, Kiela, and Cho, 2021; Schmidt, Vulić, and Glavaš, 2022, 2023). A similar issue arises in weak supervision: if the validation set contains enough cleanly annotated examples, these could theoretically be repurposed for training rather than being restricted to model selection. However, research in this area continues to rely on large numbers of high-quality validation examples solely for model selection. This reliance raises important questions about the practicality of weak supervision approaches, leading us to consider two key issues. First, in the absence of clean validation examples, can existing weakly supervised learning methods still perform effectively? Second, given that the training set contains a larger number of lower-quality examples while the validation set has fewer but higher-quality examples, would it be more effective to train directly on the clean validation data instead? In this chapter, we establish more realistic problem settings for weak supervision applications and empirically address these questions through comprehensive experiments.

The content presented in this chapter is based on:

> **Dawei Zhu**, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, Dietrich Klakow (2023). *Weaker Than You Think: A Critical Look at Weakly Supervised Learning*. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)
> URL: https://aclanthology.org/2023.acl-long.796/

5.1 INTRODUCTION

Weakly supervised learning (WSL) is one of the most popular approaches for alleviating the annotation bottleneck in machine learning. Instead of collecting expensive clean annotations, it leverages weak labels from various weak labeling sources such as heuristic rules, knowledge bases or lower-quality crowdsourcing (Ratner et al., 2017). These weak labels are inexpensive to obtain, but are often noisy and inherit biases from their sources. Deep learning models trained on such noisy data without regularization can easily overfit to the noisy labels (Tänzer, Ruder, and Rei, 2022; Zhang et al., 2017). Many ad-

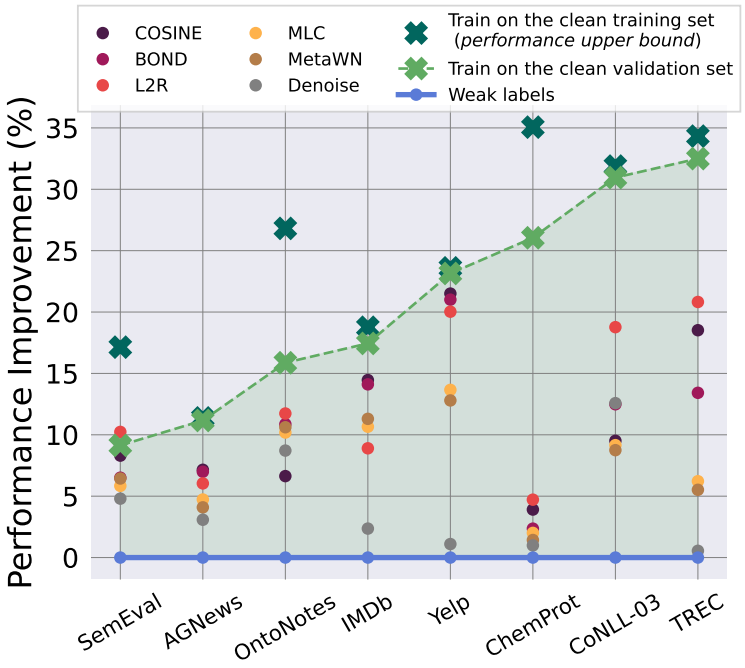


Figure 5.1: **Performance improvement over weak labels on the test sets.** Each point represents the average performance improvement of one approach over five runs. On various NLP datasets, weakly supervised methods (dots) outperform weak labels (blue line) on the test sets. However, *simply fine-tuning on the available clean validation data (light green crosses) outperforms all sophisticated weakly supervised methods in almost all cases*. See Appendix C.4.2 for experimental details.

vanced WSL techniques have recently been proposed to combat the noise in weak labels, and significant progress has been reported. On certain datasets, they even manage to match the performance of fully-supervised models (Liang et al., 2020; Ren et al., 2020; Yu et al., 2021).

In this chapter, we take a close look at the claimed advances of these WSL approaches and find that the *benefits of using them are significantly overestimated*. Although they appear to require only weak labels during training, a substantial number of clean validation samples are used for various purposes such as early-stopping (Liang et al., 2020; Yu et al., 2021) and meta-learning (Ren et al., 2018; Shu et al., 2019; Zheng, Awadallah, and Dumais, 2021). We cast doubt on this practice: in real-world applications, these clean validation samples could have instead been used for training. To address our concern, we explore fine-tuning models directly on the validation splits of eight datasets provided by the WRENCH benchmark (Zhang et al., 2021c) and compare it to recent WSL algorithms. The results are shown in Figure 5.1. Interestingly, although all WSL models generalize better than the weak labels, **simply fine-tuning on the validation splits outperforms all WSL methods in almost all cases**, sometimes even by a large

margin. This suggests that existing WSL approaches are not evaluated in a realistic setting and the claimed advances of these approaches may be overoptimistic. In order to determine the true benefits of WSL approaches in a realistic setting, we conduct extensive experiments to investigate the role of clean validation data in WSL. Our findings can be summarized as follows:

- Without access to any clean validation samples, all WSL approaches considered in this chapter *fail to* work, performing similarly to or worse than the weak labels (§5.4).
- Although increasing the amount of clean validation samples improves WSL performance (§5.5), these validation samples can be more efficiently leveraged by directly training on them, which can outperform WSL approaches when there are more than 10 samples per class for most datasets (§5.6).
- Even when enabling WSL models to continue training on clean validation samples, they can barely beat an embarrassingly simple baseline which directly fine-tunes on weak labels followed by fine-tuning on clean samples. This stays true with as few as 5 samples per class (§5.7).
- The knowledge encoded in pre-trained language models biases them to seek linguistic correlations rather than shallow rules from the weak labels; further fine-tuning the pre-trained language models with contradicting examples helps reduce biases from weak labels (§5.8).

Altogether, we show that existing WSL approaches significantly overestimate their benefits in a realistic setting. We suggest future work to (1) fully leverage the available clean samples instead of only using them for validation and (2) consider the simple baselines discussed in this chapter when comparing WSL approaches to better understand WSL's true benefits.

5.2 RELATED WORK

WEAK SUPERVISION. Weak supervision is proposed to ease the annotation bottleneck in training machine learning models. It uses weak sources to automatically annotate the data, making it possible to obtain a large amount of annotated data at a low cost. A comprehensive survey is done in Zhang et al. (2022). Ratner et al. (2017) propose to label data programmatically using heuristics such as keywords, regular expressions or knowledge bases. One drawback of weak supervision is that its annotations are noisy, i.e., some annotations are incorrect. Training models on such noisy data may result in poor generalization (Tänzer, Ruder, and Rei, 2022; Zhang et al., 2017; Zhang

et al., 2022). One option to counter the impact of wrongly labeled samples is to re-weight the impact of examples in loss computation (Ren et al., 2018; Shu et al., 2019; Zheng, Awadallah, and Dumais, 2021). Another line of research leverages the knowledge encoded in pre-trained language models (Jiang et al., 2021; Ren et al., 2020; Stephan, Kougia, and Roth, 2022). Methods such as BOND (Liang et al., 2020), ASTRA (Karamanolakis et al., 2021) and COSINE (Yu et al., 2021) apply teacher-student frameworks to train noise-robust models. Zhu et al. (2023a) show that teacher-student frameworks may still be fragile in challenging situations and propose incorporating meta-learning techniques in such cases. Multiple benchmarks are available to evaluate weak supervision systems, e.g., WRENCH (Zhang et al., 2021c), Skweak (Lison, Barnes, and Hubin, 2021), and WALNUT (Zheng et al., 2022a). In this chapter, we take representative datasets from WRENCH and reevaluate existing WSL approaches in more realistic settings.

REALISTIC EVALUATION. Certain pitfalls have been identified when evaluating machine learning models developed for low-resource situations. Earlier work in semi-supervised learning (SSL) in computer vision, for example, often trains with a few hundred training examples while retaining thousands of validation samples for model selection (Miyato et al., 2018; Tarvainen and Valpola, 2017). Oliver et al. (2018) criticize this setting and provide specific guidance for realistic SSL evaluation. Recent work in SSL has been adapted to discard the validation set and use a fixed set of hyperparameters across datasets (Li, Xiong, and Hoi, 2021; Xie et al., 2020a; Zhang et al., 2021a). In NLP, it has been shown that certain (prompt-based) few-shot learning approaches are sensitive to prompt selection which requires separate validation samples (Perez, Kiela, and Cho, 2021). This defeats the purported goal of few-shot learning, which is to achieve high performance even when collecting additional data is prohibitive. Recent few-shot learning algorithms and benchmarks have adapted to a more realistic setting in which fine-grained model selection is either skipped (Alex et al., 2021; Bragg et al., 2021; Gao, Fisch, and Chen, 2021; Lu et al., 2022; Schick and Schütze, 2022) or the number of validation samples are strictly controlled (Bragg et al., 2021; Zheng et al., 2022b). To our knowledge, no similar work exists exploring the aforementioned problems in the context of weak supervision. This motivates our work.

5.3 OVERALL SETUP

PROBLEM FORMULATION. Formally, let $\mathcal{X}$ and $\mathcal{Y}$ be the feature and label space, respectively. In standard supervised learning, we have access to a training set $D = \{(x_i, y_i)\}_{i=1}^{N}$ sampled from a clean data distribution $\mathcal{D}_c$ of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. In weak supervision, we are instead given a weakly labeled dataset $D_w =$

$\{(x_i, \hat{y}_i)\}_{i=1}^{N}$ sampled from a noisy distribution $\mathcal{D}_n$, where $\hat{y}_i$ represents labels obtained from weak labeling sources such as heuristic rules or crowd-sourcing.[1] $\hat{y}_i$ is noisy, i.e., it may be different from the ground-truth label y_i. The goal of WSL algorithms is to *obtain a model that generalizes well on $D_{test} \sim \mathcal{D}_c$ despite being trained on $D_w \sim \mathcal{D}_n$*. In recent WSL work, a set of clean samples, $D_v \sim \mathcal{D}_c$, is also often included for model selection.[2]

DATASETS. We experiment with 8 datasets covering different NLP tasks in English. Concretely, we include four text classification datasets: (1) AGNews (Zhang, Zhao, and LeCun, 2015b), (2) IMDb (Maas et al., 2011), (3) Yelp (Zhang, Zhao, and LeCun, 2015b), (4) TREC (Li and Roth, 2002), two relation classification datasets: (5) SemEval (Hendrickx et al., 2010) and (6) ChemProt (Krallinger et al., 2017), and two Named-Entity Recognition (NER) datasets: (7) CoNLL-03 (Tjong Kim Sang and De Meulder, 2003) and (8) OntoNotes (Pradhan et al., 2013). The weak annotations are obtained from WRENCH (Zhang et al., 2021c). Table 5.1 summarizes the basic statistics of the datasets.

Dataset	Task	# Class	# Train	# Val	# Test
AGNews	Topic	4	96K	12K	12K
IMDb	Sentiment	2	20K	2.5K	2.5K
Yelp	Sentiment	2	30K	3.8K	3.8K
TREC	Question	6	4,965	500	500
SemEval	Relation	9	1,749	178	600
ChemProt	Relation	10	13K	1.6K	1.6K
CoNLL-03	NER	4	14K	3.2K	3.4K
OntoNotes 5.0	NER	18	115K	5K	23K

Table 5.1: **Dataset statistics.** Additional details on datasets are provided in Appendix C.1.

WSL BASELINES. We analyze popular WSL approaches including: (1) $\mathbf{FT_W}$ represents the standard fine-tuning approach[3] (Devlin et al., 2019; Howard and Ruder, 2018). Ren et al. (2020), Zhang et al. (2021c)

1 Majority voting can be used to resolve conflicting weak labels from different labeling sources.

2 We refer to model selection as the process of finding the best set of hyperparameters via a validation set, including the optimal early-stopping time. Prior work has shown that early-stopping is crucial for learning with noisy labels (Arpit et al., 2017; Tänzer, Ruder, and Rei, 2022; Yu et al., 2021; Zhu et al., 2022).

3 We use the subscript "W" to emphasize that this fine-tuning is done on the weakly annotated data and to distinguish it from the fine-tuning experiments in Section 5.6 which are done on clean data.

and Zheng et al. (2022a) show that a PLM fine-tuned on a weakly labeled dataset often generalizes better than the weak labels synthesized by weak labeling sources. (2) **L2R** (Ren et al., 2018) uses meta-learning to determine the optimal weights for each (noisy) training sample so that the model performs best on the (clean) validation set. Although this method was originally proposed to tackle artificial label noise, we find it performs on par with or better than recent weak supervision algorithms on a range of datasets. (3) **MLC** (Zheng, Awadallah, and Dumais, 2021) uses meta-learning as well, but instead of weighting the noisy labels, it uses the meta-model to correct them. The classifier is then trained on the corrected labels. (4) **BOND** (Liang et al., 2020) is a noise-aware self-training framework designed for learning with weak annotations. (5) **COSINE** (Yu et al., 2021) underpins self-training with contrastive regularization to improve noise robustness further and achieves state-of-the-art performance on the WRENCH (Zhang et al., 2021c) benchmark.

To provide a fair comparison, we use RoBERTa-base (Liu et al., 2019) as the common backbone PLM for all WSL approaches (re)implemented in this chapter.

5.4 IS CLEAN DATA NECESSARY FOR WSL?

Recent best-performing WSL approaches rely on a clean validation set for model selection. Figure 5.1 reveals that they fail to outperform a simple model that is directly fine-tuned on the validation set. Therefore, a natural question to ask is: "*Will WSL still work without accessing the clean validation set?*". If the answer is yes, then we can truly reduce the burden of data annotation and the benefits of these WSL approaches would be undisputed. This section aims to answer this question.

SETUP. We compare three different validation choices for model selection using either (1) a clean validation set from D_v as in prior work, (2) weak labels from $\tilde{D}_v$ obtained by annotating the validation set via weak labeling sources (the same procedure used to construct training annotations), or (3) no validation data at all. In the last setting, we randomly select 5 sets of hyperparameters from our search space (see Appendix C.3). We run the WSL approaches introduced in Section 5.3 on all eight datasets with different validation choices and measure their test performance. Each experiment is repeated 5 times with different seeds.

While one may expect a certain drop in performance when switching from D_v to $\tilde{D}_v$, the absolute performance of a model does not determine the usefulness of a WSL method. We are more interested

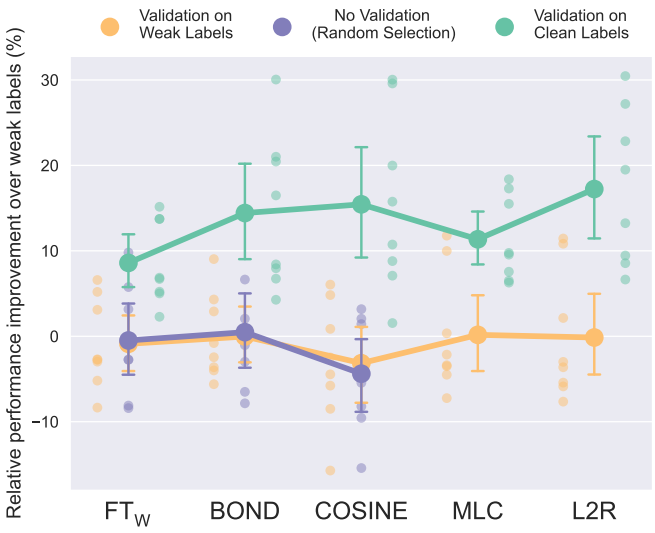
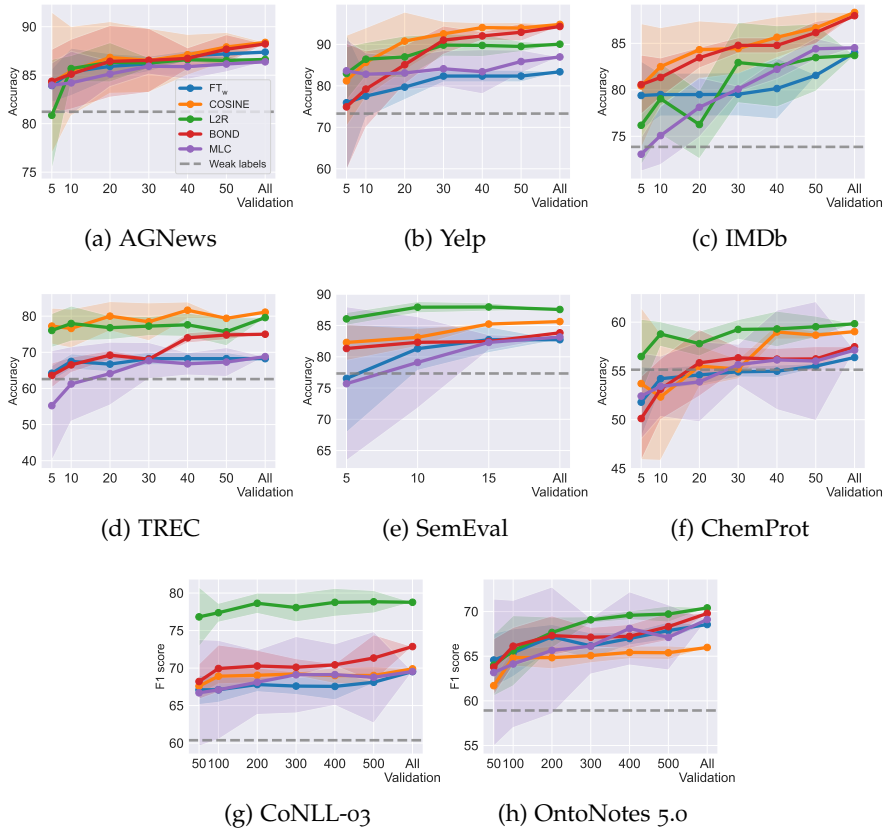


Figure 5.2: **Relative performance gain over weak labels when varying validation conditions.** The dots show the average performance gain across 5 runs for each of the 8 datasets. The curves show the average gain across datasets. WSL baselines achieve noticeable performance gains only if a clean validation set is used. Performing model selection on a weakly labeled validation set does not help generalization. Note that L2R and MLC are not applicable without validation data.

in whether a trained model generalizes better than the weak labels.[4] In realistic applications, it is only worth deploying trained models if they demonstrate clear advantages over the weak labels. Therefore, we report the relative performance gain of WSL approaches over the weak labels. Formally, let P_{WL}, P_α denote the performance (accuracy, F1-score, etc.) achieved by the weak labels and a certain WSL method α, respectively. The the relative performance gain is defined as $G_\alpha = (P_\alpha - P_{WL})/P_{WL}$. We consider a WSL approach to be *effective* and practically useful only if $G_\alpha > 0$.

RESULTS. Figure 5.2 shows the relative performance gain for all considered WSL approaches. When model selection is performed on a clean validation set (green curve), all weak supervision baselines generalize better than the weak labels. Sophisticated methods like COSINE and L2R push the performance even further. This observation is consistent with previous findings (Zhang et al., 2021c; Zheng et al., 2022a). However, when using a weakly labeled validation set (yellow curve), all WSL baselines become *ineffective* and barely outperform the

4 Weak labeling sources are typically applied to the training data to synthesize a weakly annotated training set. However, it is also possible to synthesize the weak labels for the test set following the same procedure and measure their performance. In other words, weak labeling sources can be regarded as the most basic classification model, and the synthesized weak labels are its predictions.



(a) AGNews (b) Yelp (c) IMDb

(d) TREC (e) SemEval (f) ChemProt

(g) CoNLL-03 (h) OntoNotes 5.0

Figure 5.3: **The impact of the number of clean validation samples on performance.** We plot average performance and standard deviation over 5 runs varying the size of the clean validation data. Whenever a small proportion of validation data is provided, most WSL techniques generalize better than the weak label baseline (grey dashed line). Performance improves with additional validation samples, but this tendency usually levels out with a moderate number of validation samples.

weak labels. More interestingly, models selected through the weakly labeled validation sets do not outperform models configured with random hyperparameters (purple curve). These results demonstrate that model selection on clean validation samples plays a vital role in the effectiveness of WSL methods. **Without clean validation samples, existing WSL approaches do not work.**

5.5 HOW MUCH CLEAN DATA DOES WSL NEED?

Now that we know clean samples are necessary for WSL approaches to work, a follow-up question would be: "How many clean samples do we need?" Intuitively, we expect an improvement in performance as we increase the amount of clean data, but it is unclear how quickly this improvement starts to level off, i.e., we may find that a few dozen

clean samples are enough for WSL approaches to perform model selection. The following section seeks to answer this question.

SETUP. We apply individual WSL approaches (see Section 5.3) and vary the size of clean data sub-sampled from the original validation split. For text and relation classification tasks, we draw an increasing number of clean samples $N \in \{5, 10, 15, 20, 30, 40, 50\}$ per class when applicable.[5] In the case of NER, as a sentence may contain multiple labels from different classes, selecting exactly N samples per class at random is impractical. Hence, for NER we sample $N \in \{50, 100, 200, 300, 400, 500\}$ sentences for validation. For each N, we run the same experiment 5 times. Note that the clean data is *used solely for model selection* in this set of experiments.

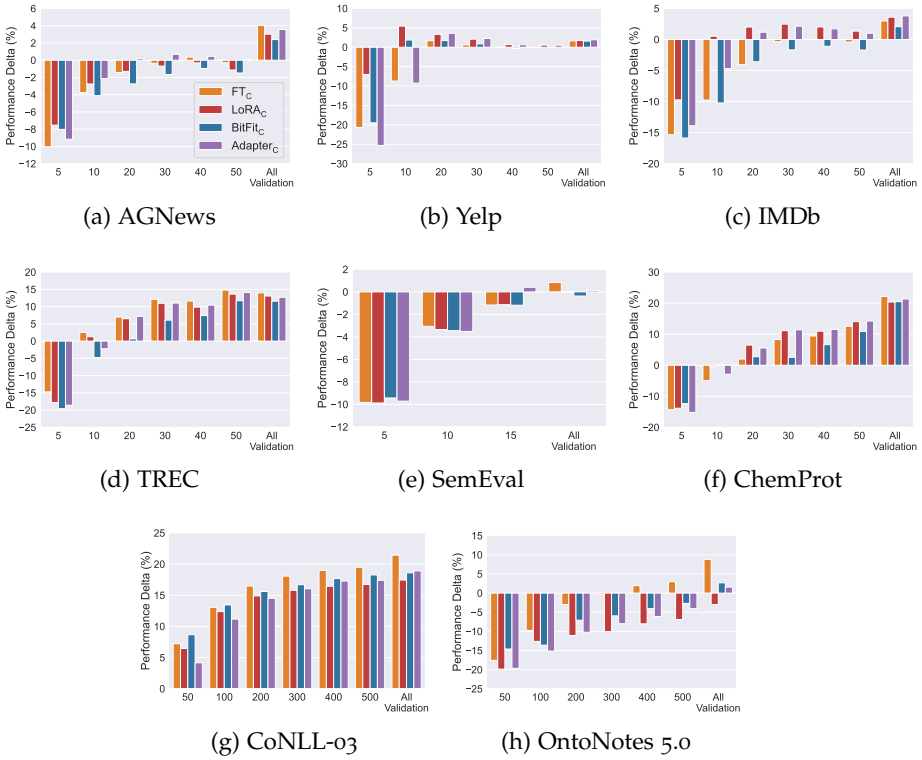
RESULTS. As shown in Figure 5.3, in most cases, a handful of validation samples already make WSL work better than the weak labels. We observe an increasing trend in performance with more validation samples, but typically this trend weakens with a moderate size of samples (~30 samples per class or ~200 sentences) and adding more samples provides little benefit. There are a few exceptions. For example, on IMDb all methods except L2R consistently perform better with more validation data. On CoNLL-03, on the other hand, most methods seem to be less sensitive to the number of samples. Overall, the results suggest that **a small amount of clean validation samples may be sufficient for current WSL methods to achieve good performance.** Using thousands of validation samples, like in the established benchmarks (Zhang et al., 2021c; Zheng et al., 2022a), is neither realistic nor necessary.

5.6 IS WSL USEFUL WITH LESS CLEAN DATA?

The previous sections have shown that current WSL approaches (1) do not improve over direct fine-tuning on the existing validation splits (Figure 5.1) and (2) require only a small amount of validation samples to be effective (Figure 5.3). This section investigates whether the conclusion from Figure 5.1 would change with less clean data, i.e., can WSL approaches outperform direct fine-tuning when less clean data is available?

SETUP. We follow the same procedure as in Section 5.5 to subsample the *cleanly annotated* validation sets and fine-tune models directly on the sampled data. In addition to the standard fine-tuning approach (Devlin et al., 2019), we also experiment with three parameter-efficient

5 The validation set of SemEval is too small to support $N > 20$. Also, if a dataset is unbalanced, we randomly select $N \times C$ samples, where C denotes the number of classes. This is a realistic sampling procedure when performing data annotation.



(a) AGNews (b) Yelp (c) IMDb

(d) TREC (e) SemEval (f) ChemProt

(g) CoNLL-03 (h) OntoNotes 5.0

Figure 5.4: **Using clean data for validation vs. training.** We show the average performance (Acc. and F1-score in %) difference between (parameter-efficient) fine-tuning approaches and COSINE when varying amounts of clean samples. COSINE uses the clean samples for validation, whereas fine-tuning approaches directly train on them (indicated in the legend with the subscript 'C'). For most sequence classification tasks, fine-tuning approaches work better once 10 clean samples are available for training. For NER, several hundreds of clean sentences may be required to attain better results via fine-tuning. Refer to Appendix C.4 for a comparison with other WSL approaches.

fine-tuning (PEFT) approaches as – in the few-shot setting – they have been shown to achieve comparable or even better performance than fine-tuning all parameters (Liu et al., 2022; Logan IV et al., 2022; Peters, Ruder, and Smith, 2019). In particular, we include adapters (Houlsby et al., 2019), LoRA (Hu et al., 2022), and BitFit (Zaken, Goldberg, and Ravfogel, 2022).

We use one fixed set of hyperparameter configurations and train models for 6000 steps on each dataset.[6] We report performance at the

6 The hyperparameters are randomly picked from the ranges mentioned in the original papers of corresponding methods and fixed across all experiments. *We did not cherry-pick them based on the test performances*. In most cases the training loss converges within 300 steps. We intentionally extend training to show that we do not rely on extra data for early-stopping. We find that overfitting to the clean data does not hurt generalization. A similar observation is made in Mosbach, Andriushchenko, and Klakow (2021). Detailed configurations are presented in Appendix C.4.

last step and compare it with WSL approaches which use the same amount of clean data for validation.

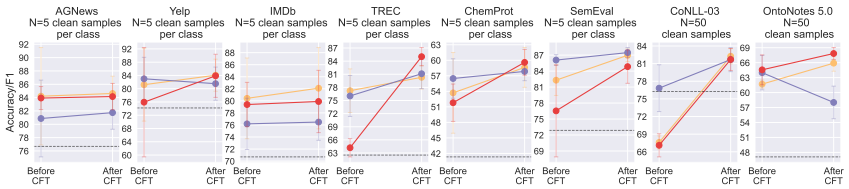
RESULTS. Figure 5.4 shows the performance difference between the fine-tuning baselines and COSINE, one of the best-performing WSL approaches, when varying the number of clean samples. It can be seen that in extremely low-resource cases (less than 5 clean samples per class), COSINE outperforms fine-tuning. However, fine-tuning approaches quickly take over when more clean samples are available. LoRA performs better than COSINE on three out of four text classification tasks with just 10 samples per class. AGNews is the only exception, where COSINE outperforms LoRA by about 1% when 20 samples per class are available, but adapters outperform COSINE in this case. Relation extraction has the same trend where 10–20 samples per class are often enough for fine-tuning approaches to catch up. For NER tasks, all fine-tuning approaches outperform COSINE with as few as 50 sentences on CoNLL-03. OntoNotes seems to be more challenging for fine-tuning and 400 sentences are required to overtake COSINE. Still, 400 sentences only account for 0.3% of the weakly labeled samples used for training COSINE. This indicates that models can benefit much more from training on a small set of clean data rather than on vast amounts of weakly labeled data. Note that the fine-tuning approaches we experiment with work out-of-the-box across NLP tasks. If one specific task is targeted, few-shot learning methods with manually designed prompts might perform even better.[7] Hence, the performance shown here should be understood as a lower bound of what one can achieve by fine-tuning. Nevertheless, we can see that even considering the lower bound of fine-tuning-based methods, **the advantage of using WSL approaches vanishes when we have as few as 10 clean samples per class**. For many real-world applications, this annotation workload may be acceptable, limiting the applicability of WSL approaches.

5.7 CAN WSL BENEFIT FROM FINE-TUNING?

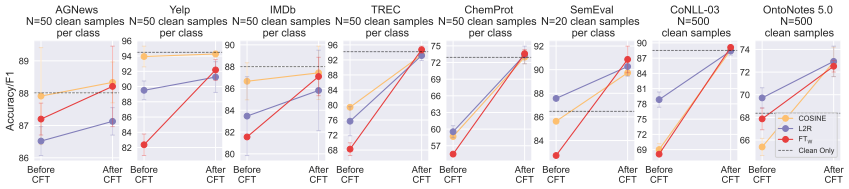
The WSL approaches have only used clean samples for validation so far, which is shown to be inefficient compared to training directly on them. We question whether enabling WSL methods to further fine-tune on these clean samples would improve their performance. In this section, we study a straightforward training approach that makes use of both clean and weak labels.[8]

7 For example, Zhao et al. (2021) achieve an accuracy of 85.9% on AGNews using just 4 labeled samples in total. For comparison, COSINE needs 20 labeled samples for validation to reach 84.21%.

8 In Appendix C.5 we also explored other baselines that combine clean and weak data, but they perform considerably worse than the approach we consider in this section.



(a) $N = 5$ clean samples per class for classification tasks. $N = 50$ clean samples for NER tasks.



(b) $N = 50$ clean samples per class for classification tasks except for SemEval due to its limited validation size. $N = 500$ clean samples for NER tasks

Figure 5.5: **Performance before and after continuous fine-tuning (CFT) on the clean data**. The average performance and standard deviation over 5 runs are reported. Though CFT improves the performance of WSL approaches in general, the simplest baseline FT_W gains the most from it. After applying CFT, FT_W performs on par with or better than more sophisticated WSL approaches, suggesting these sophisticated approaches might have overestimated their actual value. Further plots are included in Appendix C.6.

SETUP. Given both the weakly labeled training data and a small amount of clean data, we consider a simple two-phase training baseline. In the first phase, we apply WSL approaches on the weakly labeled training set, using the clean data for validation. In the second phase, we take the model trained on the weakly labeled data as a starting point and continue to train it on the clean data. We call this approach continuous fine-tuning (**CFT**). In our experiment, we apply CFT to the two best-performing WSL approaches, COSINE and L2R, along with the most basic WSL baseline, FT_W. We sample clean data in the same way as in Section 5.5. The training steps of the second phase are fixed at 6000. Each experiment is repeated 5 times with different seeds.

RESULTS. Figure 5.5 shows the model performance before and after applying CFT. It can be seen that CFT does indeed benefit WSL approaches in most cases even with very little clean data (Figure 5.5a). For L2R, however, the improvement is less obvious, and there is even a decrease on Yelp and OntoNotes. This could be because L2R uses the validation loss to reweight training samples, meaning that the value of the validation samples beyond that may only be minimal. When more clean samples are provided, CFT exhibits a greater performance gain (Figure 5.5b). It is also noticeable that CFT reduces the performance gap among all three WSL methods substantially. Even the simplest

approach, FT_W, is comparable to or beats L2R and COSINE in all tasks after applying CFT. Considering that COSINE and L2R consume far more computing resources, our findings suggest that **the net benefit of using sophisticated WSL approaches may be significantly overestimated and impractical for real-world use cases.**

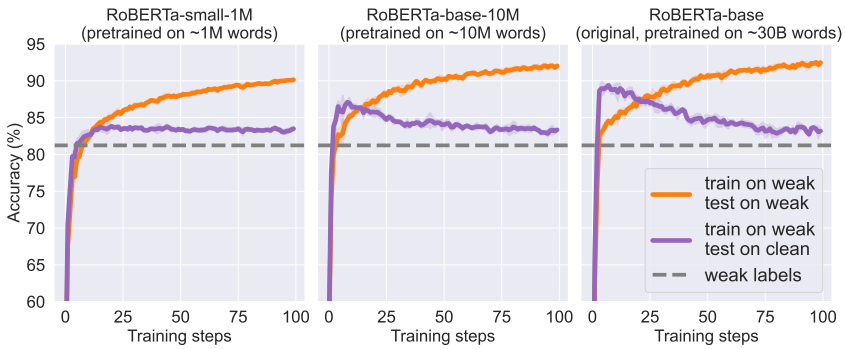
Finally, we find the advantage of performing WSL diminishes with the increase of clean samples even after considering the boost from CFT. When 50 clean samples per class (500 sentences for NER) are available, applying WSL+CFT only results in a performance boost of less than 1% on 6 out of 8 datasets, compared with the baseline which only fine-tunes on clean samples. Note that weak labels are no free lunch. Managing weak annotation resources necessitates experts who not only have linguistic expertise for annotation but also the ability to transform that knowledge into programs to automate annotations. This additional requirement naturally reduces the pool of eligible candidates and raises the cost. In this situation, annotating a certain amount of clean samples may be significantly faster and cheaper. Thus, we believe WSL has a long way to go before being truly helpful in realistic low-resource scenarios.

5.8 WHAT MAKES FT_W+CFT EFFECTIVE?

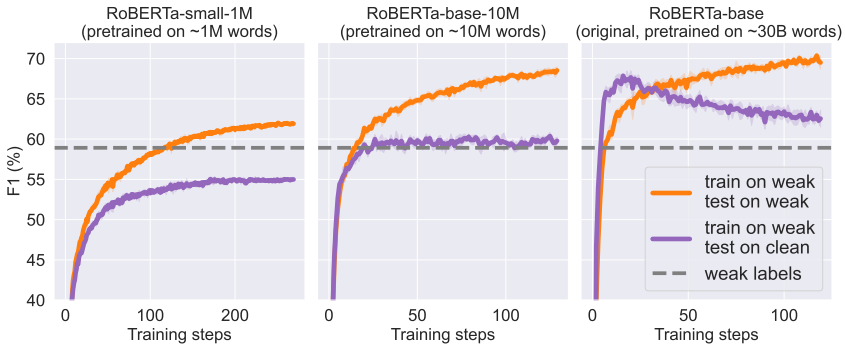
As seen in the previous section, combining FT_W with CFT yields a strong baseline that more sophisticated WSL approaches can hardly surpass. This section examines factors that contribute to the effectiveness of this method. Specifically, we aim to answer two questions: (1) "How does FT_W resist biases despite being trained only on weak labels?" and (2) "How does CFT further reduce bias introduced by weak labels?".

SETUP. To answer question (1), we modify the backbone PLM to see if its encoded knowledge plays an important role. In addition to RoBERTa-base, we explore two other PLMs that are pre-trained on less data: RoBERTa-small-1M and RoBERTa-base-10M, which are pre-trained on 1M and 10M words, respectively.[9] We report model performance on both clean labels and weak labels to see which labels the model tends to fit. To answer question (2), we vary the agreement ratio in the clean samples to see how these clean labels help combat biases from weak labels. The agreement ratio is defined as the percentage of samples whose clean labels match the corresponding weak labels. Intuitively, if the clean label for a specific training example matches its weak label, then this example may not contribute additional informa-

9 The original RoBERTa-base model is pre-trained on 100B words. The two less pre-trained models are obtained from (Warstadt et al., 2020). RoBERTa-base-10M retains the same architecture as RoBERTa-base, while RoBERTa-small-1M contains fewer parameters.



(a) AGNews

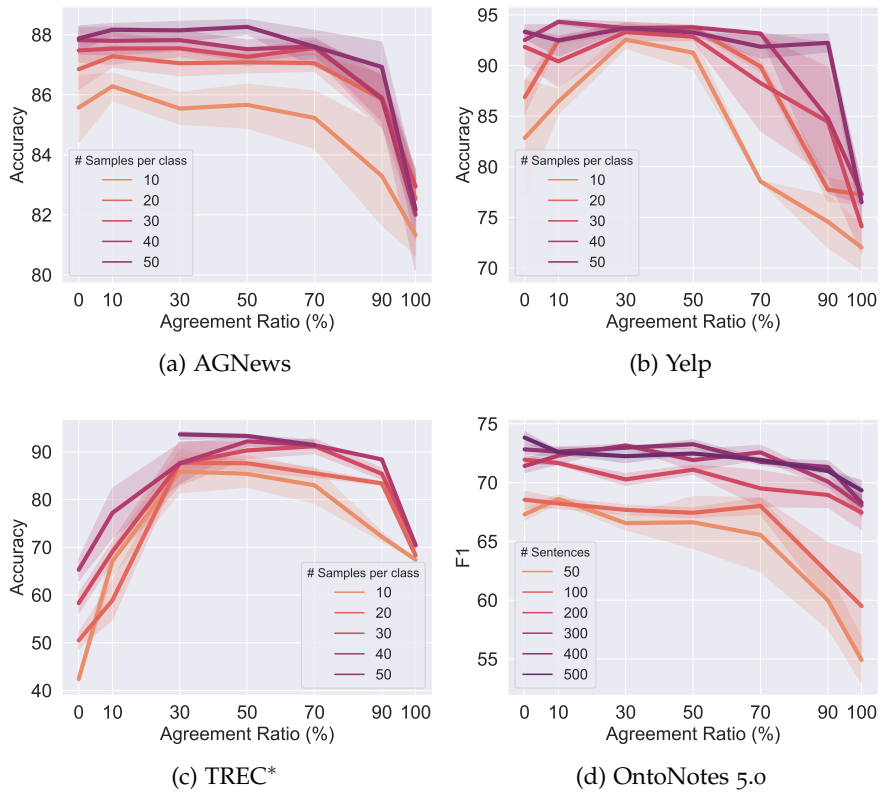


(b) OntoNotes 5.0

Figure 5.6: **Performance curves of different PLMs during training**. PLMs are trained on weak labels and evaluated on both clean and weakly labeled test sets. Pre-training on larger corpora improves performance on the clean distribution. Further plots are in Appendix C.7.

tion to help combat bias. A higher agreement ratio should therefore indicate fewer informative samples.

RESULTS. Figure 5.6 shows the performances for different PLMs. Pre-training on more data clearly helps to overcome biases from weak labels. When the pre-training corpus is small, the model tends to fit the noisy weak labels more quickly than the clean labels and struggles to outperform weak labels throughout the entire training process (figs. 5.6a and 5.6b, left). With a large pre-training corpus, however, the model can make better predictions on clean labels than weak labels in the early stages of training, even when it is only trained on weak labels (figs. 5.6a and 5.6b, right). If we apply proper early-stopping before the model is eventually dragged toward weak labels, we can attain a model that generalizes significantly better than the weak labels. This indicates that *pre-training provides the model with an inductive bias to seek more general linguistic correlations instead of superficial correlations from the weak labels*, which aligns with previous findings in Warstadt et al. (2020). This turns out to be the key to why simple FT_W works here.



(a) AGNews (b) Yelp

(c) TREC* (d) OntoNotes 5.0

Figure 5.7: **Model performance varying the number of clean samples N and agreement ratio α.** Large α generally causes a substantial drop in performance. *: Certain combinations of α and N are not feasible because the validation set lacks samples with clean and weak labels that coincide or differ. Further plots are in Appendix C.7.

Figure 5.7 shows how the agreement ratio α in clean samples affects the performance. Performance declines substantially for $\alpha > 70\%$, showing that it is necessary to have contradictory samples in order to reap the full advantage of CFT. This is reasonable, given that having examples with clean labels that coincide with their weak labels may reinforce the unintended bias learned from the weakly labeled training set. The optimal agreement ratio lies around 50%. However, having $\alpha = 0$ also yields decent performance for most datasets except TREC, suggesting contradictory samples play a more important role here and at least a minimum set of contradictory samples are required for CFT to be beneficial.

5.9 CONCLUSIONS AND RECOMMENDATIONS

Our extensive experiments provide strong evidence that recent WSL approaches heavily overestimate their performance and practicality. We demonstrated that they hinge on clean samples for model selection

to reach the claimed performance, yet models that are simply trained on these clean samples are already better. When both clean and weak labels are available, a simple baseline (FT_W+CFT) performs on par with or better than more sophisticated methods while requiring much less computation and effort for model selection.

Inspired by prior work (Oliver et al., 2018; Perez, Kiela, and Cho, 2021), our recommendations for future WSL approaches are the following:

- Report the model selection criteria for proposed methods and, especially, how much they rely on the presence of clean data.
- Report how many cleanly annotated samples are required for a few-shot learning approach to reach the performance of a proposed WSL approach. If thousands of weakly annotated samples are comparable to a handful of clean samples – as we have seen in Section 5.6 – then WSL may not be the best choice for the given low-resource setting.
- If a proposed WSL method requires extra clean data, such as for validation, then the simple FT_W+CFT baseline should be included in evaluation to claim the real benefits gained by applying the method.

We hope our findings and recommendations will spur more robust future work in WSL such that new methods are truly beneficial in realistic low-resource scenarios.

LIMITATIONS

We facilitate fair comparisons and realistic evaluations of recent WSL approaches. However, our study is not exhaustive and has the following limitations.

First, it may be possible to perform model selection by utilizing prior knowledge about the dataset. For example, if the noise ratio (the proportion of incorrect labels in the training set) is known in advance, it can be used to determine (a subset of) hyperparameters (Han et al., 2018b; Li, Socher, and Hoi, 2020). In this case, certain WSL approaches may still work without access to extra clean data.

Second, in this chapter we concentrate on tasks in English where strong PLMs are available. As we have shown in Section 5.6, training them on a small amount of data is sufficient for generalization. For low-resource languages where no PLMs are available, training may not be that effective, and WSL methods may achieve higher performance.

Third, we experiment with datasets from the established WRENCH benchmark, where the weak labels are frequently assigned by simple rules like as regular expressions (see Appendix C.2 for examples). However, in a broader context, weak supervision can have different

forms. For example, Smith et al. (2022) generates weak labels through large language models. Zhou et al. (2022) use hyper-link information as weak labels for passage retrieval. We have not extended our research to more diverse types of weak labels.

Despite the above limitations, however, we identify the pitfalls in the existing evaluation of current WSL methods and demonstrate simple yet strong baselines through comprehensive experiments on a wide range of tasks.

6

FEATURE-DEPENDENT NOISE IN MACHINE TRANSLATION

In the previous chapters, we examined noisy labels in classification tasks, primarily using BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as our learning models. However, noise can also occur in generation tasks — and in fact, it often does so more frequently. This is due to the higher annotation burden associated with generation tasks, making the cost and effort of obtaining large amounts of high-quality annotations prohibitive. Consequently, automatic data synthesis techniques are frequently used, which can result in lower-quality data (e.g., (Taori et al., 2023)). Additionally, even for human annotators, providing optimal solutions can be challenging for certain tasks. For example, annotators may struggle to "use the fewest words to summarize the text while covering all aspects of the original document."

With recent advancements in Large Language Models (LLMs), it has become common practice to unify various NLP tasks into generation formats. However, it remains unclear to what extent LLMs are resilient to noisy annotations in the training data from these generation tasks.

In this chapter, we study noisy annotations in generation tasks and use LLMs for learning. Specifically, our aim is to gain a deeper understanding of the learning behavior during Supervised fine-tuning (SFT). To maintain a focused scope for our research, we concentrate on Machine Translation (MT) as the task at hand.

Typically, SFT involves only a negligible amount of training data compared to the extensive pre-training phase of LLMs. This raises important questions: How is knowledge infused into an LLM during the SFT phase? If most knowledge is acquired during pre-training and SFT primarily serves to align model outputs with downstream tasks like MT, will the LLM remain robust to noise (errors) in the SFT data? This chapter seeks to answer these questions within the context of MT.

The content presented in this chapter is based on:

> **Dawei Zhu**, Pinzhen Chen, Miaoran Zhang, Barry Haddow, Xiaoyu Shen, Dietrich Klakow (2024). *Fine-Tuning Large Language Models to Translate: Will a Touch of Noisy Data in Misaligned Languages Suffice?*. The 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)
> URL: https://aclanthology.org/2024.emnlp-main.24/

6.1 INTRODUCTION

LLMs have reached new heights in various NLP tasks (Brown et al., 2020; Jiang et al., 2023a; Radford et al., 2019; Touvron et al., 2023b). Supervised fine-tuning (SFT, Ouyang et al., 2022, alternatively, instruction tuning or simply fine-tuning in some literature) further prepares these models for better generalization and reliability in downstream tasks by training on task input-output data combined with instructions in natural languages (Mishra et al., 2022; Sanh et al., 2022; Wei et al., 2022a). In this research direction, various works have studied the "scaling up" of SFT data size, number of languages, etc (Chung et al., 2024; Muennighoff et al., 2023). On the other hand, recent papers also embraced the philosophy of "less is more" by achieving strong results with a small set of high-quality training instances, claiming a "superficial alignment hypothesis" (Zhou et al., 2023b) with similar findings by others.

This chapter investigates the role of SFT *data* in aligning LLMs to MT, a cross-lingual generation task with high demands in practical domains. Prior research has found fine-tuning to improve translation performance (Zhang et al., 2023b) and more recent works also integrated continued pre-training with more data to provide further improvement (Alves et al., 2024; Xu et al., 2024a). For encoder-decoder models, Wu et al. (2024a) used little data to enable an English-centric model to translate between any two languages. Nonetheless, the feasibility of "less is more" in LLM translation fine-tuning is rather under-explored. In translation prompting, researchers have suggested that a model's translation capability can be attributed to the bilingual signals exposed during pre-training (Briakou, Cherry, and Foster, 2023) and task recognition in LLM layers (Sia, Mueller, and Duh, 2024), hinting that the translation capability has been picked up during pre-training. A natural question follows: *Can we put reduced effort into data?*

From a data efficiency perspective, we squeeze the translation SFT data to a mere size of 32 or the translation direction to 1 for multilingual translation, for which we believe LLMs already possess a strong pre-trained foundation in multilingual understanding and generation. Beyond quantity and language diversity, we perform SFT on synthesized data via machine translation, which is a common data augmentation practice for under-served languages. To summarize, our analysis is grounded in the task of MT, with "scaling down" in mind. In multiple dimensions—data size (§6.3.2), translation direction (§6.3.3 and §6.3.4), and data synthesis (§6.3.5)—our findings verify, complement, and refine the existing superficial alignment hypothesis for fine-tuning LLMs for translation tasks:

1. 32 data instances successfully enable an LLM to translate in 11 directions. More data still helps but the return diminishes.

2. Data in a single translation direction can effectively align an LLM to translate to and from multiple directions. Yet, it is crucial to pick the right direction—we recommend not placing English on the target side.

3. When fine-tuning on lower-quality synthetic data, LLMs are affected if the data is placed on the target side, but they show greater resilience against such flaws in low-resource languages, which are less represented during pre-training.

6.2 PRELIMINARIES

6.2.1 *Supervised fine-tuning*

In this chapter, we perform SFT to prepare pre-trained LLMs for MT. Let S denote a source input and $T = [t_1, t_2, ..., t_{|T|}]$ denote a target-side reference. We start with placing the input into a prompt template by applying $\mathcal{I}(\cdot)$ to S. For each training instance, the instruction template is randomly selected from a pre-defined pool. We fine-tune an LLM parameterized by θ by optimizing the log-likelihood:

$$\begin{aligned}\mathcal{L}_{SFT}(\mathcal{I}(S), T; \theta) &= -\log P(T|\mathcal{I}(S); \theta) \\ &= -\log \prod_{k=1}^{|T|} P(t_k|t_{<k}, \mathcal{I}(S); \theta) \\ &= -\sum_{k=1}^{|T|} \log P(t_k|t_{<k}, \mathcal{I}(S); \theta)\end{aligned}$$

6.2.2 *Superficial alignment hypothesis*

Zhou et al. (2023b) claim that a model's knowledge and capabilities are acquired almost entirely during pre-training, and the effect of alignment tuning might be "superficial", in that it teaches the model the format for interacting with users. This idea is further supported by recent works (Ghosh et al., 2024; Lin et al., 2024). However, to what extent this applies to multilingual translation in LLMs is little known. To bridge this gap, we conduct a series of controlled experiments on fine-tuning LLMs for translation, complementing previous research across three dimensions. First, we study the parallel data efficiency in the era of LLMs, aiming to determine the minimum data needed for effective model alignment to the translation task. Next, we explore the scope of alignment by probing whether aligning one translation direction influences other directions. Finally, we investigate how synthesized fine-tuning data quality impacts the LLMs' behaviour in generating translations.

6.3 EXPERIMENTS AND RESULTS

6.3.1 *Experimental setup*

TRAINING. By default, we take the test sets from WMT17 to WMT20 as our parallel training data (Barrault et al., 2019, 2020; Bojar et al., 2018, 2017); we also use the development sets in WMT21 (Akhbardeh et al., 2021) for training if a language pair of interest is not available in earlier years. The specific training data configurations will be detailed in the subsequent sections. The test sets from WMT21 are used for validation. Detailed data statistics can be found in appendix D.6.1. The LLM we use for SFT is the base version of Llama-2 7B (Touvron et al., 2023b). When performing SFT, we use a learning rate of 5e-6, an effective batch size of 64, and a linear learning rate scheduling with a warmup ratio of 0.1. We select the model checkpoint based on COMET scores on the validation sets.[1] To form the model input for SFT, we feed the source sentence into the Alpaca prompt template (Taori et al., 2023), supplementing it with a translation instruction that is randomly selected from a pool of 31 diverse instructions. Refer to Table D.2 in the appendix for a complete list of templates.

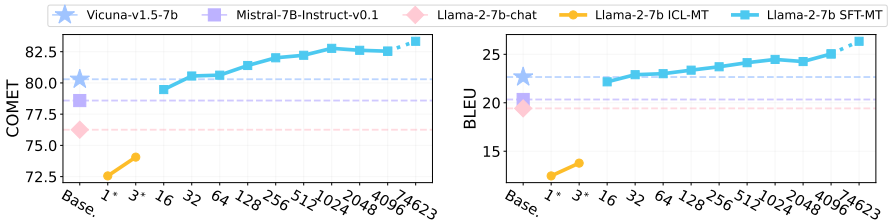


Figure 6.1: Performance comparison between instruction-tuned baselines and Llama-2 fine-tuned with different training data sizes. Average COMET (left) and BLEU (right) scores across 11 translation directions are presented. For training data sizes of 1 and 3, ICL is applied, marked with an asterisk "*"; otherwise, we perform SFT. With only 32 training examples for SFT, Llama-2 outperforms general-purpose, instruction-tuned baselines. Base.: instruction-tuned baseline models. See individual performance for the 11 translation directions in Appendix D.1.

EVALUATION. We primarily evaluate the models on the WMT22 test sets (Kocmi et al., 2022) covering 11 translation directions: en↔cs, en↔de, en↔jp, en↔ru, en↔zh, and en→hr.[2] Languages in these 11 directions are explicitly included in Llama-2's pre-training corpus.

1 In our preliminary experiments, we found that validation perplexity has a relatively weak correlation with COMET scores measured on the validation set, similar to earlier findings (Ouyang et al., 2022).

2 Language codes: cs=Czech, de=German, hr=Croatian, jp=Japanese, ru=Russian, zh=Chinese. "↔" means that both translation directions are covered. Note that only en→hr is available in WMT22 but not hr→en.

In Section 6.3.4, we extend our evaluation to translation directions involving medium and low resource languages: Icelandic and Hausa (i.e., en↔is, en↔ha), which comes from WMT21's test set. At inference time, a fixed translation instruction is applied (Table D.2 row 1). We use beam search with a beam size of 4 for generation, as our preliminary results indicate that it offers better translation quality than sampling-based generation, an observation consistent with recent works (Jiao et al., 2023a; Zeng et al., 2024). The maximum generation length is set to 256 tokens. We used a reference-based COMET22 checkpoint[3] (Rei et al., 2020) and BLEU (Papineni et al., 2002) as the evaluation metrics. See appendix D.6.3 for detailed software configurations.

6.3.2 *How much SFT data enables LLMs to translate?*

Recent works in machine translation suggest that pre-trained LLMs require significantly less parallel data for fine-tuning (via SFT), compared to training conventional translation models from scratch. However, the SFT process in these works still operates with an order of 10^5 parallel samples (Jiao et al., 2023a; Xu et al., 2024a; Zeng et al., 2024; Zhang et al., 2023b, i.a.), without a clear justification for selecting this specific data size and source. This raises a pivotal question, inspired by the recently proposed "superficial alignment hypothesis" (Zhou et al., 2023b): Is SFT mainly a method for superficially aligning LLMs for translation tasks? If so, what is the actual minimal amount of data required to achieve effective "alignment"?

SETUP. We fine-tune Llama-2 7B using different numbers of training samples and evaluate the multilingual translation performance of the resulting models. We collect training data covering 10 translation directions: en↔{cs, de, jp, ru, zh}. The training data sourced from WMT17-20 contains a total of 74,623 parallel examples. Note that the training samples across translation directions are not evenly distributed. To create training sets of varying sizes, we subsample the original data into subsets that are powers of 2, starting from 16 (2^4) and ending with 4096 (2^{12}); larger subsets always contain smaller ones. To ensure balanced language representation in our subsets, we distribute samples as evenly as possible among the language pairs.[4]

We refer to the fine-tuned model as **SFT-MT**. Considering LLMs can also perform translation through prompting, we compare SFT-MT with 1- and 3-shot in-context learning (ICL), denoted as **ICL-MT**. For ICL, we randomly select demonstrations from the training set in the test direction for each test sentence. We do not consider Llama-

3 Specifically, COMET is reported on a scale of 0 to 100 as opposed to its raw 0 to 1 range.

4 For example, the data size distribution for our 32-example training set is [4, 4, 3, 3, 3, 3, 3, 3, 3, 3].

2's zero-shot performance because, although it sometimes produces acceptable translations in the beginning, it often continues generating, which makes it difficult to accurately estimate its performance. Lastly, since LLMs fine-tuned on diverse tasks also serve as strong translation systems (Zhu et al., 2024), we compare our models with open-source general-purpose instruction-tuned LLMs, which we denote as **IT-LLM**. These include Vicuna-v1.5-7b (Chiang et al., 2023), Mistral-7b-Instruct (Jiang et al., 2023a), and Llama-2-7b-chat (Touvron et al., 2023b).[5]

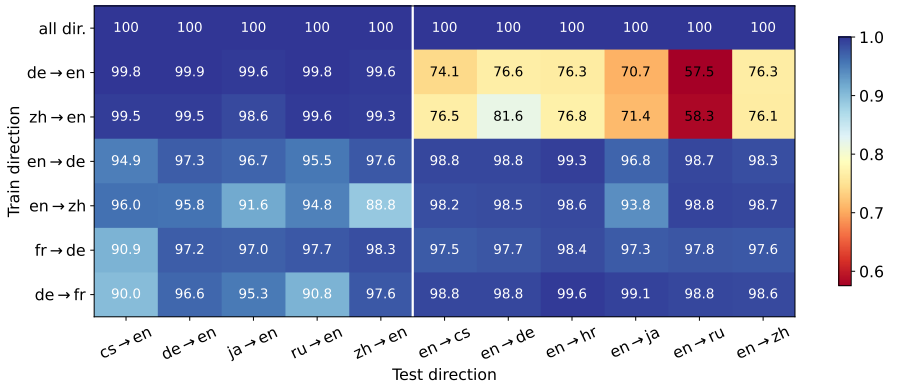


Figure 6.2: Normalized COMET score (as a % of performance from fine-tuning on an equivalent sized dataset of all 10 directions) resulted from varying combinations of train and test translation directions. In most cases, Llama-2 fine-tuned on a single translation direction can effectively translate across other directions, achieving performance comparable to models trained on all directions, with a few exceptions when trained on X→en but tested on en→X. Performance measured in BLEU score is provided in Appendix D.2.

RESULTS. Figure 6.1 illustrates the effect of varying training sizes on translation performance. In both 1- and 3-shot cases, ICL-MT underperforms IT-LLM baselines like Llama-2-7b-chat despite sharing the same foundation model, indicating that a few in-context demonstrations may not effectively align Llama-2 for translation.

However, performance significantly improves when Llama-2 is fine-tuned with just 16 samples. With further increases in the training size to 32 samples, Llama-2 performs on par with or surpasses all three IT-LLM baselines in both COMET and BLEU metrics. This suggests that a handful of high-quality parallel data can effectively specialize the model into a performant translation system. Increasing parallel data further boosts performance, though with diminishing returns: the COMET score rises by an average of 2 points when expanding from 32 to 1024 samples, but only by 0.5 points when increasing further from 1024 to 75K samples (full training set). Given that it is unlikely that these 32 training samples "teach" Llama-2 new translation skills,

5 lmsys/vicuna-7b-v1.5, Mistral-7B-Instruct-v0.1, and meta-llama/Llama-2-7b-chat-hf.

this shows strong evidence that superficial alignment applies to MT. We observe a similar trend in Mistral-7B and Llama-2-13B. Refer to Appendix D.1 for their performance across varying data sizes. In summary, effective translation alignment begins with minimal training data, revealing **less is good alignment and more is better with diminishing gains**.

6.3.3 *Do we need to include all directions?*

In the preceding section, we follow the traditional practice in multilingual MT by including multiple translation directions during training. However, the observation that only a few dozen examples make Llama-2 translate well leads us to reconsider the necessity of including samples from all directions of interest. Specifically, will training on just a single translation direction be sufficient to help LLMs perform multilingual translation?

SETUP. We explore six training configurations, each focusing on a single translation direction: de→en, zh→en, en→de, en→zh, fr→de, and de→fr. These configurations include cases where English appears on the source side, the target side, as well as settings with English excluded, to investigate if specific languages have a different impact on the overall performance. The training size is set to 1024 for SFT. Evaluations are conducted across the same 11 test directions as used in the previous section. Additionally, we explore similar settings in ICL, where we present demonstrations with translation directions that do not match those used in evaluations, to determine if the mechanisms of both SFT and ICL exhibit similarities. Lastly, we conduct a joint evaluation, progressively expanding both the training size and the range of covered translation directions to understand the combined effect of these factors.

SFT RESULTS. Figure 6.2 demonstrates the normalized performance of Llama-2 when fine-tuned in various single directions. Remarkably, training with just one direction enables Llama-2 to translate between multiple languages. For instance, after fine-tuning on de→en or zh→en, the model can translate from all considered languages to English, scoring at least 98.6% of the original COMET scores for training on all directions. Similarly, the model fine-tuned on en→de, en→zh, fr→de or de→fr also demonstrates only a slight performance decline when translating from English.

Notable declines are observed in two scenarios: (1) trained to translate to English and evaluated on translating to non-English; and (2) trained to translate to non-English and evaluated on translating to

Evaluation on de→en				
demo lang	1-shot		3-shot	
	COMET	BLEU	COMET	BLEU
de→en	73.47	19.7	75.04	22.4
en→de	55.96	7.3	44.39	3.5
de→fr	66.35	12.1	64.61	17.6
fr→de	58.06	7.8	57.13	10.5
zh→en	56.66	10.7	54.82	7.1
en→zh	51.30	7.8	56.87	1.8

Evaluation on en→de				
demo lang	1-shot		3-shot	
	COMET	BLEU	COMET	BLEU
en→de	67.37	10.5	69.80	14.3
de→en	57.83	8.7	45.54	5.0
en→zh	59.76	9.5	59.53	8.4
zh→en	47.31	4.5	49.24	5.0
fr→de	59.36	8.6	66.01	12.9
de→fr	60.70	11.0	61.76	11.3

Table 6.1: ICL-MT performance with aligned vs. misaligned demonstrations, evaluated on de→en and en→de. 1-shot/3-shot: using 1 or 3 demonstrations randomly sampled from the training set. Misaligned demonstrations consistently cause a substantial performance drop.

English.[6] Of these two scenarios, scenario 1 exhibits a much larger performance drop. The fact that both scenarios involve a mismatch between using English and non-English suggests that *Llama-2, as an English-centric LLM, may process English differently compared to other languages*. When fine-tuned for English generation, the model may misinterpret the task as only generating in English. Generalization among non-English languages is much easier than generalization between English and non-English languages, as evidenced by the negligible performance drop when fine-tuning and testing on two vastly different language pairs such as de→fr and en→zh. Overall, the findings suggest that **SFT in one translation direction effectively enables the many directions, though avoiding misinterpretation is crucial**.

6 Analysis of model outputs reveals that they often merely echo the source sentence, ignoring the translation instruction.

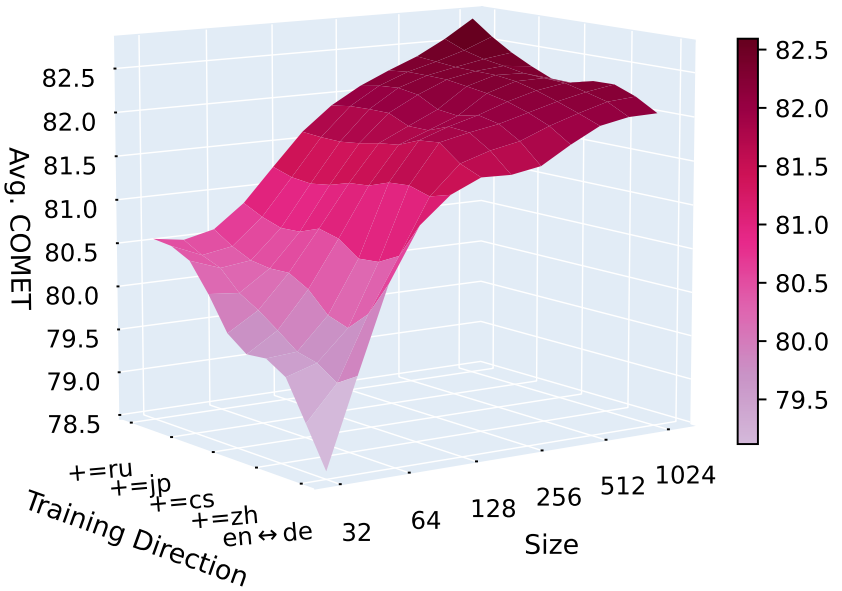


Figure 6.3: Average performance (in COMET) across 11 test directions for models trained with varying data sizes and directions. Both factors positively impact performance. +=: training directions added on top of previous directions; two directions are added at each time. For example, "+=ru" covers 10 directions: en ↔ {de, zh, cs, jp, ru}. Performance on individual test directions is provided in Appendix D.3.

ICL RESULTS. We also provide results of performing ICL with misaligned translation directions between demonstration and test in Table 6.1. It can be seen that misaligned demonstrations significantly degrade translation performance, with 3-shot be often worse than 1-shot. We observe that the model may output Chinese characters, emojis, time, etc., but no clear error patterns are observed. This contrasts sharply with findings from SFT: **while SFT can recognize the format of translation, ICL requires language-aligned demonstrations.**

JOINT EVALUATION. Figure 6.3 presents a joint evaluation of size and translation direction. For small training sizes, covering diverse translation directions in training proves to be beneficial. However, the benefits of such diversity level off as the training size increases. With a training size of 1024, models trained exclusively on two directions, en↔de, perform on par with those trained on all directions.

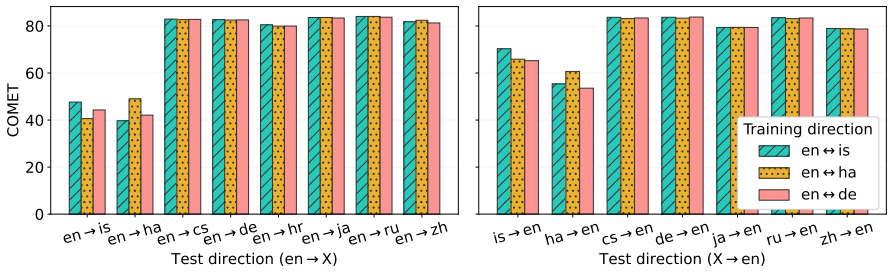


Figure 6.4: Model performance (in COMET) across 15 translation directions under different training configurations. Training models on *unseen* languages (en↔is, en↔ha) results in slight improvements in translating these languages compared to models trained on en↔de. The differences in performance when translating between *seen* languages are minimal across all training configurations. Performance measured in BLEU score is provided in Appendix D.4.

6.3.4 *Can alignment be achieved for unseen languages?*

Previous sections focus on translation directions involving languages explicitly included in Llama-2's pre-training corpus. We now extend our investigation to languages that do not have an identified presence of over 0.005% in the pre-training data (c.f. Touvron et al., 2023b, p22), referred to as *unseen* languages. Here we seek answers to two questions: (1) Can we effectively make Llama-2 translate both from and to unseen languages by fine-tuning it with a small amount of data? (2) How well can this fine-tuned model translate from and to languages *seen* in Llama?

SETUP. We consider three training configurations: en↔is, en↔ha, and en↔de, with Icelandic (is) and Hausa (ha) being unseen languages. en↔de serves as a control to assess Llama-2's initial translation capabilities into unseen languages without specific fine-tuning. The training size is fixed at 1024 (512 samples for each direction). The test directions include the 11 directions as before, plus en↔is and en↔ha coming from the WMT21 test.

RESULTS. The results are presented in Figure 6.4. It can be seen that fine-tuning on Icelandic and Hausa enhances a model's translation quality on these languages compared to the control setup, yet the gains are modest. We observe that Llama-2 manages to produce tokens in these languages, however, the translations often largely deviate from the original meanings. This suggests that it is difficult to teach models new translation directions via SFT with limited data. Interestingly, we find fine-tuning on Icelandic or Hausa does not hinder Llama-2's ability to translate from and to all seen languages, maintaining performance levels comparable to the control scenario with en↔de. Based on these results, we propose a complement to the superficial

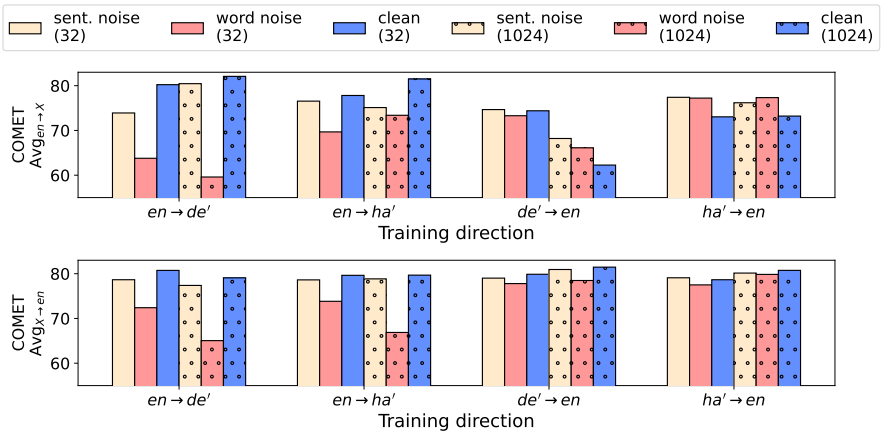


Figure 6.5: Model performance in COMET score varying training sizes, directions, and noise types. Top (Bottom): score averaged across all en→X (X→en) test directions. Training sizes considered are 32 and 1024. Generally, introducing noise on the target side tends to degrade model performance more, with the extent of impact also depending on the particular language involved. Performance measured in BLEU score is provided in Appendix D.5.

alignment hypothesis in MT: **LLMs may learn the essence of the translation task without requiring input-output mappings in languages it "understands" well.**

6.3.5 *Can we use synthesized data?*

We have observed that LLMs quickly recognize the translation task with minimal high-quality, manually curated data, but what if the quality of the training data is subpar? This situation may occur, for example when parallel data is web-crawled or machine-generated. Can LLMs still adapt to the translation task or will they overfit to the imperfections in lower-quality data, leading to degraded translation performance?

SETUP. We replace either the source or target sentences in the original training set with lower-quality synthesized ones. We try two types of data synthesis: one by translating entire sentences on the other side and another by concatenating word-to-word translations. Pleasingly, these correspond to back-translation (Sennrich, Haddow, and Birch, 2016) using translation engines or bilingual word dictionaries which are practical at different levels of resource availability. Specifically, we use the OPUS-MT suite (Tiedemann and Thottingal, 2020) to translate from English to a target non-English language.[7]

7 E.g. for de→en, the process is run in en→de with the created data reversed, hence the translated content is on the source side. Checkpoints are available on Hugging Face: `Helsinki-NLP/opus-mt-en-${trg}`.

Source	Ref./Data config.	Model output
Das finde ich ehrlich gesagt sehr ärgerlich.	reference	That really bothers me, I must say.
	literal	The find I honest said very annoying.
	en→de clean	I find that really annoying.
	en→de sent. noise	I find that honestly very annoying.
	en→de word noise	The find I honestly said very annoying.
以免再次发生这样的事情	reference	So that such a thing won't happen again.
	literal	in order to avoid again happen such thing.
	en→de clean	Let's not let it happen again.
	en→de sent. noise	In order not to happen again.
	en→de word noise	Avoid again happen this way.

Table 6.2: Examples of testing Llama-2 trained on en→de with 1024 clean and noisy target sentences. The test directions are de→en (Top) and zh→en (Bottom). The reference translation is provided by the WMT22 test set. Word-to-word references were created by the authors in consultation with native speakers. Word-level noise makes Llama-2 degenerate into a literal translator.

For word-level translation, we translate each space-delimited source word by feeding it into the MT model one at a time. Naturally, the synthesized versions introduce translation errors, adding "noise" to the training process. We investigate the impact of such noise in four translation directions: en→de′, de′→en, en→ha′, and ha′→en, where the prime (′) notation denotes the side that is created using translation (noised). We consider two training sizes: 32 and 1024. In this section, our evaluation focuses on the 11 translation directions described in section 6.3.1. Note that although Hausa is included in the current training setup, translation directions involving Hausa are excluded from our evaluation—because performance is sub-par for unseen languages as demonstrated in section 6.3.4.

RESULTS. According to Figure 6.5, it can be seen that both types of data synthesis generally cause a drop in performance. However, The degree of degradation significantly varies depending on whether the noise appears on the source or target side of the translation as well as the language. Specifically, when noise is introduced to the target side, models fine-tuned on en→de′ and en→ha′ translations exhibit a sharp decline in performance. The impact of word noise is more severe than that of sentence noise. In the case of en→de′, word-level synthesis causes the model to largely degenerate, leading to literal translations across many test cases across translation directions. An example of this behaviour is presented in Table 6.2. In contrast, the performance drop caused by word noise is less pronounced with en→ha′, particularly when evaluated on en→X.

Conversely, when noise is introduced on the source side, the negative impact is much smaller, and the disparity in performance degrada-

tion between the two types of noise diminishes. Even more strikingly, when evaluated on en→X, having noise at the source side often outperforms the clean settings. Notably, in section 6.3.3, we show that fine-tuning models purely on X→en risks task misinterpretation, leading to low performance on en→X. However, adding noise appears to mitigate this issue, resulting in improvements in both COMET and BLEU scores, especially for the ha′ →en case.

Summarizing the observations, Llama-2 is much more robust against the noise introduced in Hausa, likely because it has limited familiarity with the language, making it more difficult to detect and imitate imperfections present in the training data. As a result, Llama-2 tends to just recognize the essence of the translation task instead of overfitting to the biases present in low-quality data. In contrast, with German, Llama-2's understanding leads to a misinterpretation of the training objectives, such as fitting the word-level noise with a directive for literal translations. Overall, **LLMs may quickly fit translation imperfections in the training data, especially for seen languages; the resulting performance drop may be observable with just 32 training samples.**

6.4 RELATED WORK

6.4.1 *What does LLM SFT bring us?*

Foundational language models become more robust and follow instructions better after being fine-tuned on task-oriented supervised data formulated as natural language text (Mishra et al., 2022; Sanh et al., 2022; Wei et al., 2022a). We observe diverging trends in research on instruction tuning nowadays: (1) Many works attempt to scale up instruction data in terms of the number of tasks, languages, data size, and thus implicitly increasing training updates (Chung et al., 2024; Li et al., 2023; Muennighoff et al., 2023; Üstün et al., 2024; Wu et al., 2024c; Zhang et al., 2024a). (2) Another stream of papers, argue that instruction tuning mainly alters a base model's response style but not content or knowledge—data quality and diversity outweigh quantity (Chen et al., 2024a; Lin et al., 2024; Mitchell et al., 2024; Zhou et al., 2023b). This chapter is a continued exploration of the latter, focusing on the machine translation task. We verify the effect of size variations and include two new factors—language directions and quality—aiming to provide practical and cost-effective guidance on this matter.

Specifically, language transfer has been demonstrated in smaller pre-trained models before LLMs (Artetxe, Ruder, and Yogatama, 2020; Wu and Dredze, 2019). For (sufficiently) multilingual models, training on certain languages might still benefit other languages at the test time (Choenni, Garrette, and Shutova, 2023). In LLM instruction tuning, recent papers revealed cross-lingual transfer and improved robustness

in unseen languages via multilingual instruction tuning with a small data sample (Chen et al., 2024c; Kew, Schottmann, and Sennrich, 2023; Shaham et al., 2024). Furthermore, it has been claimed that even monolingual instruction tuning is sufficient to elicit multilingual responses in the correct languages with a key ingredient being the right learning rate (Chirkova and Nikoulina, 2024a,b). In relation to our experiments, language transfer to unseen languages might account for improved performance in language directions that are not directly fine-tuned.

6.4.2 *How can we use LLMs for translation?*

In the field of machine translation, earlier works provided analysis of general-purpose prompting (Agrawal et al., 2023; Vilar et al., 2023; Zhang, Haddow, and Birch, 2023) followed by a blossom of strategies focusing on specific aspects of the translation process (Chen et al., 2024b; Ghazvininejad, Gonen, and Zettlemoyer, 2023a; He et al., 2024; Moslem et al., 2023; Raunak et al., 2023; Sarti et al., 2023). Nonetheless, as shown in our experimental results, few-shot prompting is not on par with using instruction-tuned models, illustrating the importance of further understanding the role of instruction tuning in translation tasks.

In terms of fine-tuning LLMs for translation, previous works have explored a wide range of sub-tasks: disambiguation, low-resource, document-level, and adaptive translation, etc (Alves et al., 2023a; Iyer, Chen, and Birch, 2023; Li et al., 2024; Mao and Yu, 2024; Wu et al., 2024b; Zhang et al., 2023a). These works focus on improving translation performance and specific applications. Stap et al. (2024) show that while fine-tuning improves translation quality, it can degrade certain key LLMs' advantages, such as the contextualization ability on document-level input. Some recent research aims to enhance the translation capabilities of LLMs by incorporating human preference data (Jiao et al., 2023a; Zeng et al., 2024; Zhu et al., 2024) or by extending the pre-training phase before fine-tuning (Alves et al., 2024; Xu et al., 2024a,b), yet these approaches require significantly more data or computing resources. The aim of this chapter is not to pursue the state of the art but to investigate the opportunities of extending instruction-tuned LLMs' translation capabilities in desirable compute-efficient scenarios. It is still worth noting that our investigation is orthogonal to previous works which employ relatively large monolingual and parallel data for continued pre-training.

6.5 CONCLUSION AND FUTURE WORK

In this chapter, we conduct an in-depth analysis of fine-tuning LLMs for translation. We demonstrate that LLMs is capable of translating

in multiple directions after being fine-tuned with *minimal low-quality training data in a single direction*. While this suggests pre-trained LLMs inherently possess multilingual translation capabilities which only need to be unlocked by aligning with the correct task format, we discover pitfalls and lessons in aligning LLMs; while LLMs make efforts to adjust to the translation task, they are good at imitating other patterns such as the noise in the parallel data. Future work could explore robust training methods that align LLMs with translation while minimizing the risk of overfitting to low-quality data.

LIMITATIONS

This chapter offers a range of insights into fine-tuning LLMs for translation. However, our study is not exhaustive and is subject to the following limitations.

MODEL SIZE AND DIVERSITY. Throughout our systematic study, we fine-tuned Llama-2 7B, Llama-2 12B, and Mistral 7B. These are strong and feasible options when the work is carried out. It is important to verify the generalizability of our findings to models with different capabilities or of different sizes.

NON-ENGLISH CENTRIC MT. Our evaluation is English-centric, which is the condition of most LLM pre-training. Findings will be more comprehensive if future work can extend it to translation directions not involving English.

STATE-OF-THE-ART PERFORMANCE. Our research primarily explores how SFT enables LLM to translate to uncover data-efficient strategies in SFT and identify associated pitfalls. Recent studies have demonstrated that translation capabilities can be further enhanced through techniques such as continual pre-training (Alves et al., 2024; Xu et al., 2024a) and preference learning (Xu et al., 2024b; Zhu et al., 2024). However, these methods require significantly more training resources, which may pose challenges when applied to large models.

FINE-TUNING METHODS. Throughout this chapter, we perform SFT with full-parameter updates. It is worthwhile to explore parameter-efficient methods which bring in heavier regularization to understand whether they exhibit patterns similar to those observed in our work.

7

LEVERAGE IMPERFECT DATA IN MACHINE TRANSLATION

With data collections expanding, ensuring their quality becomes progressively more difficult. Consider Machine Translation (MT) as an example, where much of the translation data is noisy, even when provided by human translators. This noise may result from translators misinterpreting parts of the source text or prioritizing more literal translations. In this chapter, we show that even widely recognized MT test sets contain some errors, an observation also made in (Xu et al., 2024b).

Recent Large Language Models (LLMs) possess strong translation skills, so training them further on imperfect data caps their performance potential. Chapter 6 demonstrates that LLMs are prone to overfitting the noise in translation data, which leads to a performance drop. Consequently, it is difficult to significantly enhance LLMs performance solely by fine-tuning on larger datasets without a significant improvement in data quality.

Instead of consistently training LLMs with potentially noisy reference translations, we propose teaching them to distinguish between different levels of translation quality. This approach provides LLMs with a holistic understanding of translation quality across various sentences, enhancing their robustness against errors in reference translations.

Dawei Zhu, Sony Trenous, Xiaoyu Shen, Dietrich Klakow, Bill Byrne, Eva Hasler (2024). *A Preference-driven Paradigm for Enhanced Translation with Large Language Models*. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2024) URL: https://aclanthology.org/2024.naacl-long.186/

7.1 INTRODUCTION

The emergence of LLMs has significantly transformed the landscape of Natural Language Processing (NLP), showcasing outstanding capabilities in a spectrum of NLP tasks (Brown et al., 2020; Chowdhery et al., 2023; Scao et al., 2022; Touvron et al., 2023a). This transformation extends to MT (Hendy et al., 2023; Jiao et al., 2023b; OpenAI, 2023b). Through Supervised fine-tuning (SFT) using a small amount of parallel data, LLMs demonstrate the capability to compete with established commercial translation services such as Google Translate, particularly in high-resource languages (Jiao et al., 2023a; Zhang et al., 2023b).

Nevertheless, SFT trains the model to imitate reference translations token by token, making it vulnerable to the noise present within the data (Ott et al., 2018; Touvron et al., 2023b; Zhou et al., 2023a). The noise can stem not only from the lack of attention by annotators, but also from the inherent challenge of achieving perfect translations due to the intricate interplay of language, culture, and vocabulary. As an adept translator requires not only linguistic proficiency but also a deep understanding of cultural contexts and nuances in both the source and target, it is nearly unattainable to gather extensive parallel translations of top-notch quality (Herold et al., 2022; Khayrallah and Koehn, 2018; Maillard et al., 2023). As a result, the performance enhancement achieved through SFT often quickly reaches a plateau. Further increasing the volume of parallel translations typically yields minimal additional benefits, and may instead impair the translation capabilities of LLMs (Xu et al., 2023).

To alleviate aforementioned limitation of SFT, endeavors have been made to provide LLMs with holistic assessment of contrasting examples rather than token-level imitations. Chen et al. (2023) and Jiao et al. (2023a) add a flawed translation to the reference translation in the model input, encouraging the target LLM to recognize their quality difference. Zeng et al. (2023) also use a pair of translations, but they additionally optimize the LLM to favor better translations through ranking loss. Nevertheless, these works have shared limitations. First, the flawed translations are either generated by adding artificial noise to the reference translations or by other (smaller) MT systems. These imperfections in translations can be obvious and easy for LLM to distinguish, weakening the learning signal. Second, they only provide the relative ranking of the two translations, without quantifying the extent of their quality differences.

In this chapter, we present a framework based on the Plackett-Luce model to explicitly align the generation probability of the target LLM with human preferences (Plackett, 1975). Instead of using artificial noise, we collect contrasting translations generated by our target LLM, directing our optimization efforts toward "hard negative examples" (Robinson et al., 2021). Human preferences are denoted with precise scores rather than general ranking orders to teach LLMs about the nuances in different translations. LLMs are then trained to enhance their capabilities incrementally from the learnt nuances without depending solely on the existence of "gold references", so as to effectively break the plateau associated with SFT.

We build a dataset, which we refer to as MAPLE, to facilitate preference learning. It equips each source sentence with five translations in diverse quality, scored by professional translators. By performing preference learning on MAPLE, our final MT model outperforms other MT models based on the same foundation LLM by up to 3.96 COMET score. We further show that while the intention of creating MAPLE

is to enhance our target LLM, it can be reused to improve other LLMs, helping them break the performance plateau with up to 1.4M parallel data. Finally, we analyze the key factors that make preference learning effective.

Our contributions are as follows. **(1)** We leverage preference learning to teach LLMs a holistic notion of translation quality. Extensive experiments show that our model consistently outperforms strong baselines on two test sets across four translation directions. **(2)** We revisit the underlying modelling assumptions leading to the Bradley-Terry and Plackett-Luce ranking models and discuss how preference distances can be incorporated directly into the ranking models. **(3)** We meticulously construct an MT-oriented preference dataset, MAPLE, employing professional human translators to obtain quality scores for multiple translations corresponding to the same source sentence. We release our dataset to facilitate future MT research. **(4)** Our in-depth analysis reveals that high-contrast pairs and accurate quality scores are crucial in enhancing the effectiveness of our approach, providing guidance for maximizing the benefits of preference learning.

7.2 RELATED WORK

LLM-BASED MT. One simple and effective approach to use LLMs for translation tasks is through prompting. Research in this field involves examining the impact of model sizes, the number of examples ("shots") used, and template choices (Bawden and Yvon, 2023; Mu et al., 2023; Zhang, Haddow, and Birch, 2023; Zhang et al., 2024c). Moreover, (Ghazvininejad, Gonen, and Zettlemoyer, 2023b; He et al., 2023) highlight that better translations can be achieved by adding supplementary information to prompts, or engaging LLMs in related tasks prior to translation. Alternatively, another research direction seeks to fully tailor LLMs for MT tasks. Alves et al. (2023b), Chen et al. (2023), Jiao et al. (2023a), Zeng et al. (2023), and Zhang et al. (2023b) further train LLMs on parallel data via (parameter-efficient) fine-tuning. Xu et al. (2023) show that increasing the size of parallel data may not further improve LLM. The diminished returns from increasing data volume are likely due to data noise. Recent analyses suggest that quality trumps quantity when it comes to data effectiveness (Zhou et al., 2023a; Zhu et al., 2023b). Leveraging these insights, we goes beyond merely fitting the reference translations. Instead, we aim to enhance the LLM's ability to discern translations of varying quality, encouraging the generation of more precise translations while suppressing flawed outputs.

HUMAN PREFERENCE ALIGNMENT. Ouyang et al. (2022) align LLMs with human intentions and values by training a reward model for preference ranking and optimizing the LLMs through the PPO

algorithm (Schulman et al., 2017). However, the online reinforcement learning nature of PPO leads to considerable computational costs and is known for its sensitivity to hyperparameters (Huang et al., 2022; Islam et al., 2017). To ease the alignment, Dong et al. (2023) and Hu et al. (2023) suggest offline RL algorithms where samples are pre-generated. Further research goes a step beyond by directly employing the target LLMs as reward models. Yuan et al. (2023) use a ranking loss to steer LLMs towards generating helpful responses and avoiding harmful ones. In a similar vein, Hejna et al. (2023), Rafailov et al. (2023), and Song et al. (2023) use the Plackett-Luce model (Plackett, 1975) to capture human preferences in alignment. In this chapter, we adopt the Plackett-Luce model to MT, teaching the model to discern nuances in different translations and to prefer accurate translations.

7.3 METHODOLOGY

We aim to enhance LLM in MT tasks via a two-stage optimization process. We first fine-tune the target LLM with a small set of high-quality parallel data to elicit its translation ability (Section 7.3.1). This mirrors the supervised fine-tuning approach used in prior work, where LLMs were tailored to follow instructions (Taori et al., 2023; Zheng et al., 2023). We then use preference learning to guide the LLM to prioritize the generation of accurate translations over flawed ones (Section 7.3.2).

7.3.1 *Supervised fine-tuning*

We begin with optimizing our target LLM on parallel data to specialize it for translation. Let x and y denote the source and target sentence, respectively. Following Jiao et al. (2023a) we first construct a prompt by applying an instruction template $\mathcal{I}$ to x. The instruction template is randomly sampled from an instruction pool for each training sample. The target LLM, denoted by π_θ is optimized through the log-likelihood loss:

$$\begin{aligned}\mathcal{L}_{SFT}(\pi_\theta) &= -\log \pi_\theta(x, y) \\ &= -\sum_t \log P_{\pi_\theta}(y_t | y_{1,\cdots,t-1}, \mathcal{I}(x))\end{aligned} \tag{7.1}$$

where $\pi_\theta(x, y)$ denotes the likelihood of π_θ generating output y given input x. Note that in a standard implementation, a decoder-only LLM will also predict tokens within $\mathcal{I}(x)$, we zero-out the loss on these tokens as our main goal is to teach translation, not to model the input distribution (Touvron et al., 2023b).[1]

1 As per Ouyang et al. (2022), we use the term "SFT" which is interchangeably referred to as "instruction-tuning" or simply "fine-tuning" in current literature to convey the same concept.

7.3.2 *Preference learning*

The goal of the preference learning stage is to explicitly optimize the target LLM to favor accurate translations over erroneous ones. Formally, consider a set of translations $y^1, \cdots, y^L$ corresponding to a source sentence x. We assume that these translations are ordered by preference: $y^i \succ_x y^j$ for $i < j$. That is, translation y^i is preferred over y^j as a translation of the source sentence x. We further assume that there is some underlying reward model r^* that reflects the quality of the translations, which we cannot access but which we can approximate. Under the Plackett-Luce ranking model (Plackett, 1975), the distribution of preferences can be formulated as follows:

$$p^*(y^{1:L}_{\succ_x}|x) = \prod_{i=1}^{L-1} \frac{\exp(r^*(x, y^i))}{\sum_{j=i}^{L} \exp(r^*(x, y^j))} \tag{7.2}$$

where $y^{1:L}_{\succ_x}$ is a shorthand for the complete preference ranking $y^1 \succ_x \cdots \succ_x y^L$. In practice, given a training set $\mathcal{D}$ with translations equipped with a preference ranking, a reward model r_θ can be trained via maximum likelihood estimation (Cheng, Dembczynski, and Hüllermeier, 2010):

$$\mathcal{L}_{PL}(r_\theta) = -\mathbb{E}_{x, y^{1:L}_{\succ_x} \in \mathcal{D}} \sum_{i=1}^{L-1} \Big[r_\theta(x, y^i) - \log \sum_{j=i}^{L} \exp(r_\theta(x, y^j)) \Big] \tag{7.3}$$

Following recent work (Hejna et al., 2023; Rafailov et al., 2023; Song et al., 2023), we parameterize the reward model using the target LLM π_θ and rewrite the above objective as:

$$\mathcal{L}_{PL}(\pi_\theta) = -\mathbb{E}_{x, y^{1:L}_{\succ_x} \in \mathcal{D}} \sum_{i=1}^{L-1} \log \frac{\pi_\theta(x, y^i)}{\sum_{j=i}^{L} \pi_\theta(x, y^j)} \tag{7.4}$$

where $r_\theta := \log(\pi_\theta)$. Through optimizing Equation 7.4, we explicitly align the LLM generation probability with the translation quality.

A caveat when optimizing Equation 7.4 is that the ranking information omits any measure of absolute translation quality, which may lead to inadvertent suppression of the likelihood of good translations. Consider a case where we have a pair of translations, y^1 and y^2, which are both acceptable translations but have different word orders that causes minor difference in preference. Optimizing Equation 7.4 may cause the model to raise the probability of y^1 and to suppress the

probability y^2, which may damage the model.[2] To address this issue, we follow Song et al. (2023) to consider the preference distance in $\mathcal{L}_{PL}$:

$$\mathcal{L}_{PLD}(\pi_\theta) = - \mathbb{E}_{x,y^{1:L}_{\succ x} \in \mathcal{D}} \sum_{i=1}^{L-1} \log \frac{\pi_\theta^{d_i^i}(x, y^i)}{\sum_{j=i}^{L} \pi_\theta^{d_i^j}(x, y^j)}$$

where

$$d_i^j = r^*(x, y^i) - r^*(x, y^j), \text{for } j > i$$
$$d_i^i = \max_{j>i}(d_i^j) \tag{7.5}$$

We obtain the ground truth preference value $r^*(x, y)$ through human annotation, which will be detailed in Section 7.4. Finally, we combine a SFT loss calculated on the best translation y^1 with $\mathcal{L}_{PLD}$, making the complete loss function:

$$\mathcal{L} = \mathcal{L}_{PLD} + \beta \mathcal{L}_{SFT} \tag{7.6}$$

where the hyperparameter β balances the strengths of preference learning and SFT. We use **PL** as an abbreviation of our preference learning method (i.e., optimizing Equation 7.6) in the subsequent text.

We now provide some justification for directly incorporating preference distances into the Plackett-Luce model by studying the original derivation of the binary case ($L = 2$) (Bradley, 1953; Hamilton, Tawn, and Firth, 2023; Mosteller, 1951; Thurstone, 1927). Denote the preferences for y^i and y^j by random variables X_i and X_j such that the probability that y^i is preferred to y^j is $\pi_{ij} = P(X_i > X_j)$. Assuming that X_i and X_j follow Gumbel distributions[3] with locations s_i and s_j and a common scale parameter γ, the difference between the two random variables $d_{ij} = X_i - X_j$ follows a logistic distribution with location $s_i - s_j$ and scale γ:

$$d_{ij} \sim \frac{1}{4\gamma} \operatorname{sech}^2\left(\frac{d_{ij} - (s_i - s_j)}{2\gamma}\right) \tag{7.7}$$

2 Cheng and Hüllermeier (2008) show that, while the preference can be learned asymptotically solely through ranking information, incorporating additional, more detailed, preference information (e.g., distance) makes the learning process more data-efficient. Table 7.6 presents an ablation study.

3 Assuming preferences arise from a large number of i.i.d. contributions, a normal distribution results in the limit if these are averaged while the Gumbel distribution results from taking their maximum (Hamilton, Tawn, and Firth, 2023).

By defining $\pi_i = e^{s_i}$, it follows that

$$\begin{aligned}\pi_{ij} &= P(d_{ij} > 0) \\ &= \int_0^\infty \frac{1}{4\gamma} \operatorname{sech}^2\left(\frac{d_{ij} - (s_i - s_j)}{2\gamma}\right) d d_{ij} \\ &= \frac{\pi_i^{\frac{1}{\gamma}}}{\pi_i^{\frac{1}{\gamma}} + \pi_j^{\frac{1}{\gamma}}}\end{aligned} \tag{7.8}$$

Usually the scale parameter γ is set to 1 which yields the Bradley-Terry model (Bradley and Terry, 1952) (and Equation 13 of Bradley (1953)).

To introduce distance information for the binary preference case, we first note that $d_1^1 = d_1^2$ for $L = 2$ (from Equation 7.5). We then take $\gamma = \frac{1}{d_1^2}$ and $\pi_i = \pi_\theta(x, y^i)$, which yields:

$$\pi_{12} = \frac{\pi_\theta^{d_1^2}(x, y^1)}{\pi_\theta^{d_1^1}(x, y^1) + \pi_\theta^{d_1^2}(x, y^2)} \tag{7.9}$$

This shows that, for the binary case, preference distances based on the ground truth preferences can be incorporated exactly into the Bradley-Terry distribution by assuming that the X_1 and X_2 have Gumbel distributions with location parameters $s_i = \log \pi_\theta(x, y^i)$ and scale parameter $\gamma = \frac{1}{r^*(x,y^1) - r^*(x,y^2)}$.

We derive and discuss the more general case of Equation 7.5 ($L > 2$) in Appendix E.1.

CONNECTIONS WITH DPO The preference learning framework investigated here shares a common origin with DPO (Rafailov et al., 2023) in the Bradley-Terry and Plackett-Luce models over rankings (Equation 7.2, and Equation 18 of Rafailov et al. (2023)). Here, the target LLM π_θ serves directly as the reward function ($r_\theta = \log(\pi_\theta)$), whereas the DPO reward function also includes a reference distribution π_{ref} that arises from the KL-divergence constraint term in its RL objective function. By contrast, regularization in this chapter is through an external SFT term (Equation 7.6) distinct from the reward function. We note also that the use of distance functions based on ground truth reference values brings additional information into our ranking model beyond preference order alone.

7.4 HUMAN PREFERENCE DATA COLLECTION

We build MAPLE (**MA**chine translation dataset for **P**reference **LE**arning), a dataset derived from WMT20/21 test sets. It contains multiple translations per source sentence, each assigned a real-valued human preference score. MAPLE covers four translation directions: German-to-English (de→en), Chinese-to-English (zh→en), English-to-German

(en→de), and English-to-Chinese (en→zh). For each direction, 1.1K source sentences are sampled from the test sets of WMT20/21. Each source sentence is associated with five translations, including one reference translation from WMT20/21, and four translations generated by VicunaMT, our target LLM that we aim to improve through preference learning (see training details of VicunaMT in Section 7.5.1). Among the four translations, one is generated using beam search with a beam size of four, and three translations are obtained through nucleus sampling (Holtzman et al., 2020) with $p = 0.9$. We also build a development set containing 200 source sentences per direction sourced from News Crawl 2022. Altogether, MAPLE contains 5.2K source sentences and 26K translations with preference scores. See Appendix E.2.1 for more detail on the translation collecting process.

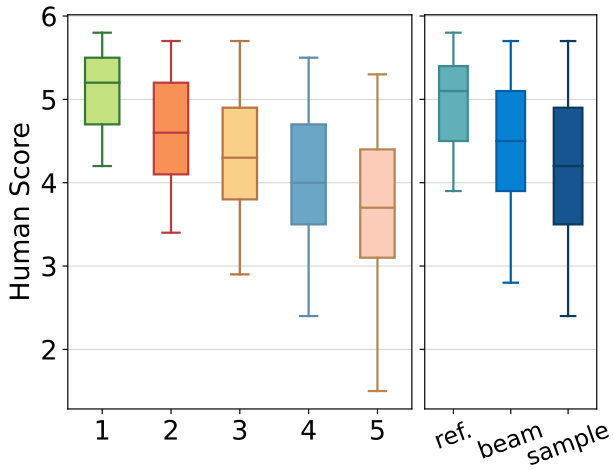


Figure 7.1: Human score distribution of translations by rank (left) and source (right).

ANNOTATION GUIDANCE. We send both the source sentence and the corresponding five translations to a panel of translators for evaluation. Each example (source sentence and its translations) is assigned to two different professional translators. They observe the source and the five translations at the same time, and assign scores between 1 (worst) and 6 (best) in increments of 0.2 using a slider. See Appendix E.2.2 for the full scoring rubric.

DATASET STATISTICS. The score distribution is shown in Fig. 7.1. The left side shows the score distribution by rank, and we can see MAPLE contains translations that exhibit a wide range of qualities. The right side shows the score distribution by translation type, and as expected the reference is ranked highest, followed by the beam search and the nucleus samples. Nonetheless, there is considerable overlap in the score distributions, and we find that in 21% of the cases, the beam search predictions are scored higher than the reference translation.

Source	Zu einem großen Tuning-Treffen ist es am Samstagabend (25. Juli 2020) in Nürnberger Südstadt gekommen. *(A large tuning meeting took place on Saturday evening (July 25, 2020) in Nuremberg's Südstadt district.)*
Reference translation	A large tuning meetup took place in a city south of Nürnberg this Saturday evening.
Best translation	On Saturday evening (25th July 2020) a large tuning meeting took place in Nuremberg's south district.

Table 7.1: An example where the reference translation is less accurate than the best model prediction. More examples are in Appendix E.2.4.

Table 7.1 shows an example where the reference translation contains an error.

7.5 EXPERIMENTS

In this section, we present our MT model trained using the proposed two-stage framework and compare it with strong LLM-based MT systems.

DATASETS. We train and evaluate the model on data on four translation directions: en↔de and en↔zh. In the SFT stage, we use high-quality test sets from WMT17/18/19 for training, containing 30K parallel sentences in total across the four directions. The WMT21 test set is used for validation. In the preference learning stage, we train on MAPLE, and validation is done on the remaining data from WMT20/21 test sets which was not selected for inclusion in MAPLE. We evaluate trained models on the test sets of WMT22 (Kocmi et al., 2022) and FLORES-200 (Costa-jussà et al., 2022). Refer to Appendix E.3.1 for detailed data statistics.

TRAINING. In both SFT and PL stages, we use a learning rate of 5e-6, an effective batch size of 96, and a linear learning rate schedule with a warmup ratio of 0.1. For each training instance, one MT instruction is randomly selected from an instruction pool containing 31 MT instructions. See Appendix E.3.2 for the complete list of instructions.

EVALUATION. At inference time, a fixed MT translation instruction is used. The maximum generation length is set to 512. We use a beam

size of 4 for decoding and report BLEU (Papineni et al., 2002) and COMET (Rei et al., 2022) scores.

7.5.1 *SFT makes good translation models*

The SFT stage seeks to train a well-performing foundation MT model using parallel data. When applying SFT, we can either select a pre-trained LLM, or its instruction-tuned version. Prior research uses both types of LLMs interchangeably, leaving it unclear which is preferable in practice. To address this gap, we explore three popular families of open-access LLMs, performing SFT on both their raw (i.e., only pre-trained) and instructed-tuned versions. Specifically, we consider LLaMA-1 (Touvron et al., 2023a), Mistral (Jiang et al., 2023b) and BLOOM (Scao et al., 2022); and their instruction-tuned versions, which are Vicuna (Zheng et al., 2023), Mistral-Instruct, and BLOOMZ (Muennighoff et al., 2023). The 7B parameter variants of these models are used here.

RESULTS. Table 7.2 presents the results before and after SFT. It can be seen that LLMs without instruction-tuning, e.g., BLOOM, perform poorly; we observe that they tend to overgenerate and repeat tokens in the source sentences.[4] In contrast, instruction-tuned models work out-of-the-box and exhibit decent performance. It can be also observed that SFT dramatically boosts the performance of raw LLMs, and slightly benefits instruction-tuned LLMs. For BLOOM and Mistral, the performance gap between raw and instruction-tuned models is mostly lost after SFT. An interesting case is Vicuna, where there is a considerable improvement on en↔zh over its base model LLaMA-1. This implies that instruction-tuned LLMs may serve as a better base model for SFT. In addition, different LLMs excel in diverse translation directions and their instruction-tuned versions do not deviate from this pattern. For example, both BLOOM and BLOOMZ perform quite well on en→zh, but have a deficiency in en→de. For LLaMA-based models, the opposite holds. This could be due to the fact that German and Chinese are not included (at least, not intentionally) in BLOOM's and LLaMA's pre-training corpora, respectively.

The Vicuna+SFT model has the best overall performance and so we select it as our target LLM to be improved through preference learning. We call this model **VicunaMT**. The generated translations in the MAPLE dataset are produced by this model.

4 Overgeneration is also noticed in (Bawden and Yvon, 2023), while it can be partially alleviated by prompt engineering and text post-processing (Srivastava et al., 2023), enhancing LLMs' zero-shot performance is not our primary focus.

	de→en	en→de	en→zh	zh→en	Avg.
			WMT22		
BLOOM	49.86	41.95	51.59	55.21	49.65
+*SFT*	77.21	69.17	84.60	78.76	77.44
BLOOMZ	74.58	62.52	83.10	78.29	74.62
+*SFT*	77.24	69.32	**84.95**	**78.77**	77.57
Mistral	54.18	49.08	49.10	55.47	51.96
+*SFT*	83.15	81.10	81.48	78.05	80.95
Mistral-Ins.	82.45	80.39	76.57	77.73	79.28
+*SFT*	82.68	81.23	82.49	77.73	81.03
LLaMA-1	63.29	55.29	45.80	55.17	54.89
+*SFT*	83.30	82.54	77.58	75.78	79.80
Vicuna	82.55	82.02	81.42	74.81	80.20
+*SFT*	**83.55**	**82.79**	81.27	77.39	**81.25**
			FLORES-200		
BLOOM	55.03	42.36	53.82	60.25	52.86
+*SFT*	83.69	67.43	**86.06**	**85.45**	80.66
Mistral	42.36	32.74	33.35	42.10	37.64
+*SFT*	88.63	84.49	80.97	85.17	84.81
Mistral-Ins.	88.04	82.55	73.20	83.70	81.87
+*SFT*	88.21	83.73	82.41	84.77	84.78
LLaMA-1	58.89	52.71	42.77	49.92	51.07
+*SFT*	88.50	84.82	76.73	83.09	83.29
Vicuna	87.82	84.17	81.52	81.53	83.76
+*SFT*	**88.66**	**86.27**	80.62	84.44	**85.00**

Table 7.2: Model performance (in COMET score) before and after performing SFT on parallel data. Rows in blue indicate instruction-tuned LLMs. Best results are in **bold**. Instruction-tuned LLMs yield high COMET scores even without SFT. Raw LLMs benefit the most from SFT. Vicuna performs the best on average on both test sets. We exclude BLOOMZ on FLORES-200 as it is a part of BLOOMZ's training data. Performance measured by BLEU score is reported in Appendix E.4.

7.5.2 *Refining through preference learning*

BASELINES. We continue training our VicunaMT model on MAPLE through preference learning and compare it with the following competitive systems from recent work: (1) **ParroT** (Jiao et al., 2023a) adds a "Hint" field to the model input, prompting the model to generate both correct and incorrect translations. At inference time, the "correct" version of the translations is used for evaluation. (2) **TIM** (Zeng et al., 2023) incorporates standard SFT with a ranking loss computed on a pair of correct and incorrect translations. (3) **SWIE** (Chen et al., 2023) proposes to attach an instruction adapter to enhance LLMs' long-term attention for better translation. (4) **ALMA** (Xu et al., 2023) first continues pre-training the LLM on monolingual data, followed by performing SFT on parallel data. Furthermore, as the preference learning stage

System	WMT22					FLORES-200				
	de→en	en→de	en→zh	zh→en	Avg.	de→en	en→de	en→zh	zh→en	Avg.
Commercial LLMs & LLaMA-2-7B based MT systems										
ChatGPT$_{(3.5\text{-turbo-}0613)}$	85.38	86.92	87.00	82.42	85.43	89.58	88.68	88.56	86.91	88.02
GPT-4$_{(gpt\text{-}4\text{-}0613)}$	85.57	87.36	87.29	82.88	85.78	89.66	88.89	88.91	87.25	88.68
ALMA-7B$_{(LLaMA\text{-}2)}$	83.98	85.59	85.05	79.73	83.59	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$
BLOOMZ-mt-7B based LLMs										
ParroT$_{(BLOOMZ\text{-}mt)}$	78.00	73.60	83.50	79.00	78.53	-*	-*	-*	-*	-*
TIM$_{(BLOOMZ\text{-}mt)}$	77.65	74.16	84.89	79.50	79.05	-*	-*	-*	-*	-*
SWIE$_{(BLOOMZ\text{-}mt)}$	78.80	75.17	84.53	79.15	79.41	-*	-*	-*	-*	-*
LLaMA-1-7B based LLMs										
ParroT$_{(LLaMA\text{-}1)}$	82.40	81.60	80.30	75.90	80.05	88.40	84.60	81.20	83.40	84.40
TIM$_{(LLaMA\text{-}1)}$	82.80	82.32	80.03	75.46	80.15	88.08	85.00	80.93	83.18	84.30
SWIE$_{(LLaMA\text{-}1)}$	82.97	81.89	80.14	76.14	80.29	88.39	85.21	81.14	83.50	84.56
VicunaMT$_{(LLaMA\text{-}1)}$	83.55	82.79	81.27	77.39	81.25	88.66	86.27	80.62	84.44	85.00
+ REF	83.88	83.37	82.86	78.19	82.07	88.48	86.11	83.35	84.54	85.62
+ BEST	83.61	83.08	83.20	78.35	82.06	88.67	85.87	84.02	84.55	85.78
+ PL	**84.23**	**84.43**	**84.26**	**79.07**	**83.00**	**88.83**	**86.73**	**84.88**	**84.76**	**86.30**

Table 7.3: Model performance in COMET scores. Best results of LLaMA-1 based models are in **bold**. Applying prefrence learning (+PL) on top of our VicunaMT model consistently leads to improvements in all cases, achieving the highest average performance among all BLOOM and LLaMA-1 based MT models. Performance in BLEU scores is reported in Appendix E.5. $^{\otimes}$: LLaMA-2 based models were not evaluated due to license constraints. WMT22 results are extracted from the original paper. *: BLOOMZ-family models use FLORES-200 for training.

introduces additional data, a performance gain could be trivial by exposing the model with more samples. To establish a fair comparison, we design two additional baselines: (5) **REF** trains VicunaMT with the reference translations in MAPLE. (6) **BEST** trains VicunaMT with the translations that are scored highest by our annotators. See Table 7.1 for an example comparison of the reference and best translations. All aforementioned baselines are performed on 7B LLMs (based either on BLOOM-7B or LLaMA-7B). Finally, we also compare our model against commercial LLMs, including ChatGPT and GPT-4.

RESULTS. We report the MT performance of various baselines in Table 7.3. It can be seen that our VicunaMT model performs well compared to recent MT systems. PL further increases the performance advantage. Our final model, VicunaMT+PL, achieves the highest average performance (83 on WMT22 and 86.3 on FLORES-200), consistently outperforming all LLaMA-1 based models across all directions, with the largest improvement being a 3.96 increase in COMET score. (en→zh on WMT22). Notably, LLaMA-based models are originally much weaker in directions involving Chinese. Through preference learning, VicunaMT reaches a translation performance close to BLOOM-based LLMs. This becomes practically significant when the goal is to deploy a single LLM to handle multiple translation directions.

Also, the PL model scores higher than VicunaMT models fine-tuned on the reference and best translations, indicating that the performance gain does not just come from having more data. Compared to the ALMA model, which is based on LLaMA-2 (Touvron et al., 2023b), a widely recognized superior open access LLM, our model demonstrates only a slight deficit of 0.59 COMET scores. Note that our strategy is orthogonal to ALMA's approach, which leverages monolingual data. Combining both strategies should lead to even better performance.

We supplement our assessment with a human evaluation, contrasting VicunaMT+PL with SFT-only Vicuna variations including VicunaMT and VicunaMT+REF, as illustrated in Table 7.4. The human evaluation confirms the trend observed with automatic metrics, where PL substantially outperforms SFT-only variations.

	de→en	en→de	en→zh	zh→en
		VicunaMT+PL vs.		
VicunaMT	+3.7%	+4.4%	+5.6%	+5.7%
VicunaMT+REF	+3.7%	+2.5%	+5.0%	+3.5%

Table 7.4: Relative improvements of VicunaMT+PL over SFT-only models (VicunaMT and VicunaMT+REF), assessed through human evaluation on the WMT22 test set, employing the same scoring criteria as those specified in MAPLE. A two-sided t-test was conducted, with 95% confidence intervals noted as ±1.7%. Positive values indicate the improvement achieved by VicunaMT+PL compared to the other models.

7.6 ANALYSIS

REUSE OF PREFERENCE DATA. MAPLE contains the translations generated by VicunaMT, which is also the target LLM we aim to improve. There would be additional value if this data could be reused to improve other LLMs. To investigate this, we train both Mistral-Instruct and BLOOMZ on MAPLE using PL. As shown in Table 7.5, PL improves both models, suggesting that the MAPLE is not limited for use with VicunaMT and can be reused for improving other LLMs.

LIMITED GAINS WITH ADDITIONAL PARALLEL DATA. Section 7.5.2 shows that the MAPLE dataset, which contains 4.4K preference examples, can be more valuable than an equivalent amount of parallel data with either the reference or the best translations. A natural follow-up question is whether adding more parallel data can close the gap. To answer this question, we collect more data by concatenating WMT20, WMT21 test data with News Commentary v16, making 1.4M parallel

	WMT22				
	de→en	en→de	en→zh	zh→en	Avg.
BLOOMZ†	77.24	69.32	84.95	78.77	77.57
+REF	77.41	68.47	84.76	79.50	77.53
+BEST	77.48	68.64	85.15	79.59	77.72
+PL	**77.83**	**69.84**	**85.36**	**80.67**	**78.42**
Mistral-Ins.†	82.68	81.23	82.49	77.73	81.03
+REF	83.06	82.63	83.39	78.07	81.79
+BEST	82.98	81.84	83.34	78.33	81.62
+PL	**83.35**	**82.94**	**84.71**	**79.25**	**82.56**

Table 7.5: Model performance in COMET scores. Best results are in **bold**. MAPLE can be reused to improve BLOOMZ and Mistral-Instruct. See results on FLORES-200 and in BLEU scores in Appendix E.6.†: SFT stage has already been applied to these models.

sentences in total.[5] We fine-tune VicunaMT and Mistral-InstructMT (i.e., Mistral-Instruct after SFT stage) on different proportions of this data and plot the performance curve in Figure 7.2. In both cases, similar to observations in (Xu et al., 2023), adding more parallel data does not always improve these models and they never attain the performance level reached by using PL with MAPLE.

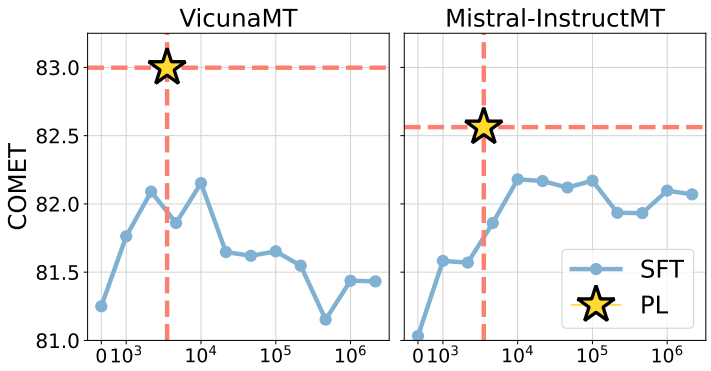


Figure 7.2: Performance comparison between PL using 4.4K examples from MAPLE and SFT, employing up to 1.4M parallel data. Evaluation is done on WMT22, and COMET scores are averaged across four translation directions. Performing SFT on more parallel data does not always lead to performance gain. PL consistently outperforms SFT in all cases.

DIVERSE TRANSLATIONS HELP MORE. By default, we perform PL using all five translations provided by MAPLE. We now study the relation between the final model performance and the number of preference translations used. We select $K = \{2, 3, 4\}$ translations and rerun the PL algorithm on VicunaMT and Mistral-InstructMT. We

5 We select News Commentary for its high-quality, domain-matching parallel data to WMT test data. WMT20/21 are included as MAPLE is built on a subset from them.

explore two selection modes for selecting K translations. Given five translations sorted by human preference scores in descending order, the *forward* mode selects the first K translations (i.e., the best K), while the *reverse* mode select the first and last $K-1$ translations. We compare both modes varying K and present the results in Figure 7.3. There is a clear disparity in performance with these two selection modes. The reverse mode consistently outperforms the forward mode given the same number of translations, with a larger advantage in low-resource cases, such as when $K = 2$. This is intuitive since the reverse mode always includes the highest- and lowest-scored translations and thus, PL may have a better chance to see "hard negatives" which have low human preference score but high generation probability. The general trend shows that including more preference samples is better, and using all available samples yields the best performance.

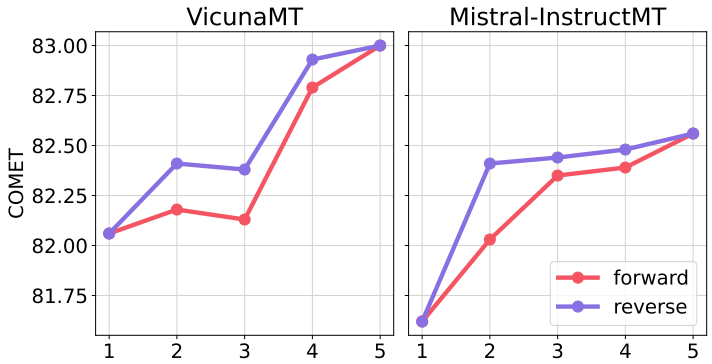


Figure 7.3: Model performance varying number of translations (K) per source sentence. Evaluation conducted on WMT22 and COMET scores averaged across four translation directions are reported. Reverse mode selects more diverse translations and achieves better performance, especially when fewer translations are provided.

	VicunaMT	Mistral-InstrctMT
SFT stage	81.25	81.03
PL stage	83.00	82.56
w/o $\mathcal{L}_{SFT}$	83.00	82.54
w/o distance	82.22	81.92
w/o $\mathcal{L}_{SFT}$/dist.	74.65	60.70
$\mathcal{L}_{SFT}$ only	82.07	81.79

Table 7.6: Ablation study. PL is less sensitive to $\mathcal{L}_{SFT}$ than the distance information. Disabling both factors leads to substantial model degradation.

DISTANCE INFORMATION IS CRUCIAL. Our framework considers the distance information in preference scores (Equation 7.5). We now investigate if this information can be replaced by simply using

the ranking information. That is, we set $d_i^j = 1$ for all translations and rerun the PL algorithm. Table 7.6 shows that when the distance information is available, excluding the SFT loss does not harm the performance much. In fact, we achieve the best performance when setting $\beta = 0$ for VicunaMT. However, when the distance information is withheld, we see a clear degradation in performance. We find that a larger β value is required when relying only on the ranking information, but this makes the PL algorithm closer to SFT. As a result, when only the ranking information is provided, VicunaMT performs similarly to the $\mathcal{L}_{SFT}$ only baseline. Finally, disabling both L_{SFT} and distance cause a large performance drop.

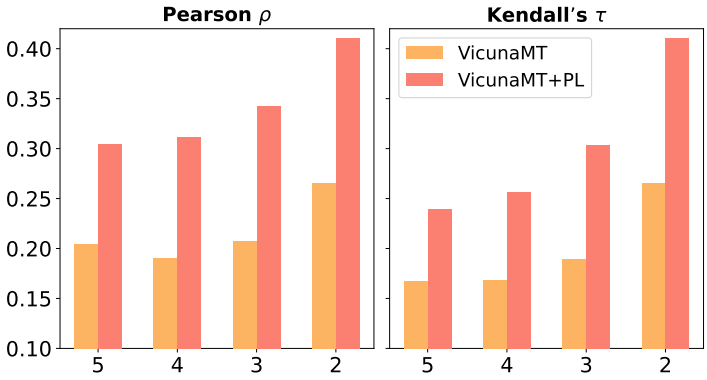


Figure 7.4: Sentence-level correlation between model generation probability and human preference scores varying number of translations (K). PL helps the model align better with human judgement.

BETTER MODEL CALIBRATION. In our preference learning framework, the model learns both translation and the ability to differentiate between different translation quality. We analyze if PL has successfully transferred human preference to the model. Using the held-out set of MAPLE, we examine the sentence-level correlation between the scores assigned by the human annotators and model generation probability. Specifically, we compute the average Pearson and Kendall's tau correlation varying the number of preference samples (reverse mode). The results are presented in Figure 7.4. Compared to the SFT baseline, VicunaMT, PL substantially improves the correlation, suggesting that our final model aligns better with human preference.

7.7 CONCLUSION

We present a preference learning framework to break the performance plateau faced when performing SFT. It enhances the translation capabilities of LLMs by motivating them to differentiate the nuances in different translations. To support this framework, we have carefully curated a preference dataset, named MAPLE, featuring translations

of varying quality, each scored by professional translators. Extensive experiments, including human evaluations, confirm the effectiveness of this framework. In addition, we demonstrate that MAPLE can be reused to enhance other LLMs, further bolstering its practical usability. Future research could consider extending our framework into an iterative process for continuous improvement of LLMs' translation capabilities.

8

CONCLUSIONS AND FUTURE PROSPECTS

Over the past decade, DNN–based AI systems have undergone rapid advancements, with numerous model architectures and training algorithms being proposed. These innovations have continuously emerged and evolved, challenging and reshaping our understanding of Natural Language Processing (NLP). In this dynamic field, Learning with Noisy Labels (LNL) remains an indispensable topic and is growing in importance. It has been consistently observed that the quality of data plays a pivotal role in determining the performance of LLMs. However, the immense volume of data required to train these models makes manual verification intractable. Consequently, incorporating some flawed data into the training process becomes unavoidable. Therefore, enhancing our comprehension of these data imperfections and their consequences is crucial for successfully training high-performing models, particularly at large scales.

8.1 SUMMARY OF THE CONTRIBUTIONS

This thesis presents a series of research studies on LNL, progressing from classification tasks to generation tasks and transitioning from PLMs to LLMs. We address different machine learning scenarios that involve noisy training sets. Through our research, we uncover insightful findings on how flawed annotations affect model performance. Building on these insights, we developed effective approaches for handling noise. Specifically, this thesis concludes with the following contributions:

- In Chapter 3, we demonstrate that PLMs exhibit remarkable robustness against feature-independent noise, even at high noise ratios. Although noise memorization eventually occurs, PLMs initially fit the clean data distribution more quickly, likely due to their better alignment with pre-trained knowledge. Building on this observation, we propose using early-stopping as a noise-handling strategy. Despite its simplicity, early-stopping effectively enables the model to generalize well, often matching or surpassing more complex noise-handling methods that require additional hyperparameter tuning and computational resources. Furthermore, these complex methods need to be retuned when factors such as data distribution, noise type, or noise levels change, whereas our approach remains unaffected by such variations. We found that existing noise-handling methods primarily slow down the noise memorization process without significantly

improving peak generalization performance during training. In contrast, early-stopping is sufficient to halt training precisely at the model's highest performance point, capturing the optimal state without requiring additional adjustments.

- We show that feature-dependent noise requires different handling approaches compared to feature-independent noise and is often more challenging to address. In Chapter 4, we propose a meta-learning-based self-training approach. This method teaches a student model using filtered pseudo-labels provided by a teacher network. Evaluated on multiple classification tasks from the WRENCH (Zhang et al., 2021c) benchmark, our approach achieves state-of-the-art performance, demonstrating its effectiveness in managing feature-dependent noise.

- We observe that many existing noise-handling approaches assume access to clean validation data. In Chapter 5, we investigate whether this assumption can be relaxed or discarded. We find that all the approaches we examined failed because: a) hyper-parameter settings obtained from a noisy validation set do not work, and b) early-stopping is ineffective on a noisy validation set under feature-dependent noise, which is often required by these methods. Consequently, we conclude that a certain amount of high-quality validation data is necessary to effectively handle feature-dependent noise. To address this, we propose a simple approach. First, we train on the noisy training data using the clean validation set for early-stopping. Then, we fine-tune the model on the clean validation set. We show that this approach is effective across different settings and highly competitive with other common noise-handling methods. Notably, the second learning phase with fine-tuning on the clean validation set does not require a second round of early-stopping. We observed that PLMs tend not to overfit on the clean data even when trained for a long time, a similar observation made by Mosbach, Andriushchenko, and Klakow (2021).

- We also investigate the impact of noise in generation tasks, with a focus on machine translation. In Chapter 6, we demonstrate that performing Supervised fine-tuning (SFT) with 32 high-quality parallel data points enables pre-trained LLMs to function as effective translation systems for language pairs well-represented in their training data. Increasing the dataset size to 70K yields only marginal improvements, suggesting that LLMs inherently possess translation capabilities and that SFT primarily serves to align these models with the specific task format. However, data quality plays a critical role: incorporating lower-quality parallel data during SFT significantly reduces performance, likely because it causes LLMs to misinterpret the task (e.g., producing inconsistent

or "waggly" translations). Consequently, avoiding low-quality data in SFT is crucial for ensuring better generalization in LLMs. At the same time, maintaining exclusively flawless translation examples in SFT datasets becomes increasingly challenging at scale, as even professional translators occasionally make errors. To address this issue, in Chapter 7, we introduce a preference learning approach that trains LLMs to differentiate between translations of varying quality. This method not only improves generation performance but also outperforms approaches relying solely on SFT, implicitly mitigating the effects of noise in the SFT data.

8.2 FUTURE DIRECTIONS

It has been repeatedly emphasized that LLMs require high-quality data for both pre-training and post-training stages (Chen et al., 2024a; Guo et al., 2025; Liu et al., 2024a; Touvron et al., 2023b; Zhou et al., 2023b, i.a.). However, given the massive data size requirements, manually inspecting all data points is infeasible. Currently, automatic data filtering is often used to remove low-quality and unsafe content. These filtering methods are often rule-based (e.g., n-gram coverage ratio (Rae et al., 2021)), suggesting that exploring the use of small DNNs to enhance filtering results could be beneficial. For example, MarcoLLM (Ming et al., 2024) utilizes similarity scores from LASER embeddings for parallel data to filter out sentence pairs with low scores. This approach is more accurate than traditional methods, such as comparing the lengths of source and target sentences. A promising direction is to explore different filtering models or methods tailored to specific text domains or to implement hierarchical filtering approaches.

While incorporating DNNs may slow down filtering, it is not necessary to clean all training data at once. Instead, one can iteratively build a higher-quality data pool by filtering out lower-quality samples. This "annealing data" (Grattafiori et al., 2024) is used in the final pre-training stage and significantly boosts performance. Notably, this annealing strategy resembles the first-noisy-then-clean approach from Chapter 5.

Another promising direction is synthesizing high-quality data. Many recent datasets and benchmarks use synthesized data (Asai et al., 2023; Guo et al., 2025; Tang and Yang, 2024; Taori et al., 2023). For instance, Guo et al. (2025) demonstrate that strong models, such as DeepSeek-R1 Zero, can be trained without any human annotations for reasoning paths in the post-training phase, since these paths can be automatically generated when guided by reliable, reward-hacking–proof reward functions. In this context, ensuring the use of robust reward functions to prevent noise in the reinforcement learning process or developing noise-tolerant reinforcement learning approaches represents a promising avenue for future research.

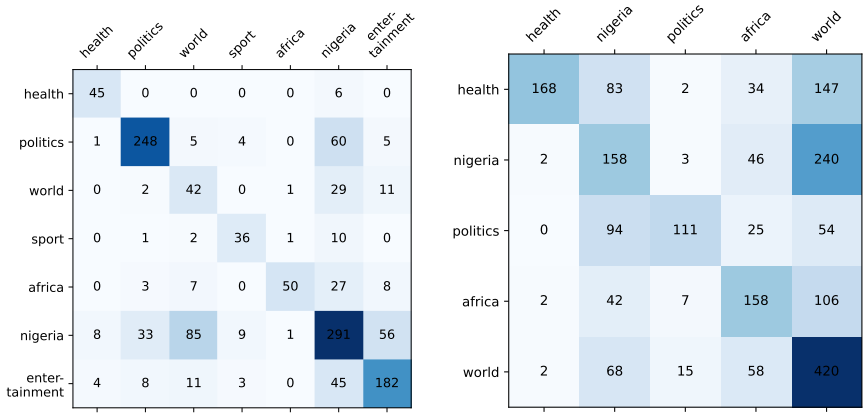
APPENDIX

A

HANDLING FEATURE-INDEPENDENT NOISE

A.1 NOISE MATRIX ON YORÙBÁ AND HAUSA

The training and validation sets of Yorùbá and Hausa have two sets of labels: the human-annotated (clean) labels and labels obtained from weak supervision. This makes it possible to compute the ground truth noise matrix in the training set. The noise matrices in Yorùbá and Hausa are shown in Figure A.1. The Noise Matrix method evaluated in Section 3.4 uses these two matrices for initialization. The labeling rules in the weak supervision are described in (Hedderich et al., 2020). The Yorùbá dataset has a rather low noise level, and the diagonally-dominant noise assumption holds in the training set. Oppositely, the Hausa training set is quite noisy. For the label "nigeria" the wrong labels is overwhelming, violating the diagonally-dominant noise assumption. Label "politics" is often misrecognized as "nigeria". Moreover, many labels are misrecognized as the label "world", making an unbalanced classification dataset. These factors make it very challenging to conquer the noise in this dataset.



(a) Yorùbá (b) Hausa

Figure A.1: Noise matrix constructed from the Yorùbá (Hausa) training set.

A.2 COMPARING EARLY-STOPPING ON CLEAN AND NOISY VALIDATION SETS

We compare the difference in model performance when using a noisy validation set rather than the clean one. Table A.1 presents the results on datasets with injected noise. For a noise level below 60% uniform

Dataset	Noise Type	Percentage	Performance Difference (%)
AG-News	uniform	40%	0.10±0.09
		60%	0.56±0.50
		70%	1.96±0.97
	single-flip	20%	0.06±0.07
		40%	0.29±0.19
		45%	2.00±0.60
IMDB	single-flip	20%	0.14±0.19
		40%	1.71±2.05
		45%	1.76±2.79

Table A.1: Average performance difference (in %) and standard deviation (5 trials) between the test accuracy based on early-stopping with the clean validation and with the noisy validation set.

	Yorùbá		Hausa	
	FT	TP+FT	FT	TP+FT
Performance Difference (%)	1.93±1.71	1.00±0.70	1.08±0.79	1.92±1.64

Table A.2: Average difference (in %) and standard deviation (10 trials) between the test accuracy based on early-stopping on the clean validation and on the noisy validation set.

noise or 40% single flip-noise, we see the difference is often less than 0.5%, indicating that a noisy validation set can already serve as a good estimator for the generalization error. In an even higher noise level, the difference can be up to 2.14%. As for the datasets obtained from weak supervision, the difference is higher in general. Table A.2 summarizes the difference on the Yorùbá and Hausa.

A.3 BERT PERFORMANCE ON DIFFERENT DATASETS AND NOISE SETTINGS

We evaluate the baselines under different noise settings and different datasets. The full result is shown in Table A.3 and Table A.4. A visualization of the result on AG-News with single-flip noise can be found in Figure A.2 (other plots can be found in the main paper). BERT clearly shows its robustness against injected noise. Although noise-handling methods do help under a high noise level, the effect is limited (less than 4%). Compared to injected noise, the noise from weak supervision is much more challenging for BERT, especially on the Hasua dataset. For both noise types, there is no single noise-handling method

	AG-News							IMDB			
		uniform			single-flip				single-flip		
	clean	40%	60%	70%	20%	40%	45%	clean	20%	40%	45%
NV	94.07±0.13	84.48±0.78	61.61±3.18	43.78±5.07	90.46±0.37	76.06±0.33	64.74±0.94	94.03±0.13	86.34±0.77	65.05±0.90	58.97±1.26
CT	-	92.18±0.21	89.90±0.38	84.74±2.56	93.33±0.12	90.62±0.53	87.99±1.64	-	92.32±0.27	89.36±0.67	83.77±3.88
NMat	-	92.25±0.14	89.91±0.48	83.9±1.87	93.91±0.15	93.13±0.31	92.93 ±0.51	-	92.07±0.21	87.13±0.44	78.82±1.37
NMwR	93.64±0.06	92.02±0.20	89.91±0.33	84.77±2.24	93.03±0.17	90.23±0.65	88.93±0.68	93.68±0.14	92.12±0.35	85.94±0.86	80.17±2.57
LS	94.43±0.19	92.45±0.21	89.79±0.38	86.64±0.78	93.56±0.23	92.40±0.33	90.94±0.86	94.06 ±0.09	92.13±0.43	87.22±1.39	80.61±2.48
WN	94.40±0.13	92.40±0.25	89.53±0.75	85.49±0.76	93.80±0.08	92.33±0.35	88.94±0.92	93.98±0.15	92.13±0.21	85.88±2.78	80.12±4.09

Table A.3: Average test accuracy (%) and standard deviation (5 trials) on AG-News and IMDB with uniform and single-flip noise. NV: without noise-handling and no validation set, i.e. train the model without noise-handling and until the training loss converges. CT: Co-teaching. NMat: Noise Matrix. NMwR: Noise Matrix with Regularization. LS: Label Smoothing. CT and NMat are equivalent to WN in the clean setting. Note that as IMDB is a binary-classification task, single-flip noise is equivalent to the uniform noise in this case.

	Yorùbá		Hausa	
	clean	noisy	clean	noisy
NV	74.11±0.26	63.88±1.59	83.02±0.45	46.98±1.01
CT	-	61.37±1.58	-	31.65±2.71
NMat	-	65.96±0.81	-	46.58±0.88
NWwR	73.78±0.32	61.32±0.71	83.21±0.40	35.36±3.60
LS	74.22±0.37	65.44±1.67	83.44±0.35	46.44±0.78
WN	74.45±0.32	64.72±1.45	83.55±0.47	46.97 ±0.81

Table A.4: Average test accuracy (%) and standard deviation (10 trials) on Yorùbá and Hausa with noise from weak supervision. NV: without noise-handling and no validation set, i.e. train the model without noise-handling and until the training loss converges. CT: Co-teaching. NMat: Noise Matrix. NMwR: Noise Matrix with Regularization. LS: Label Smoothing.

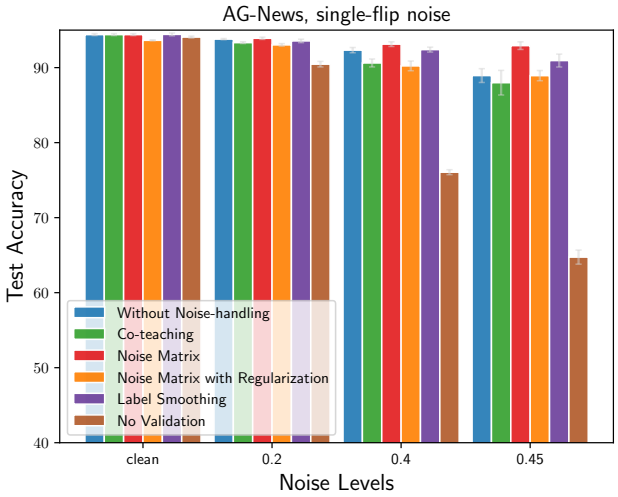


Figure A.2: Test accuracy (%) on AG-News dataset with single-flip noise.

that outperforms the simple baseline method without noise-handling in all settings.

A.4 MORE ROC CURVES

We present additional ROC curves under different settings with injected noise in Figure A.3. It is obvious that the AUC decreases when the noise levels increase. However, the absolute AUC score remains at a high level even under extremely high noise levels of injected noise.

A.5 IMPLEMENTATION DETAILS

DATASETS We experiment with the following four datasets: AG-News, IMDB, Yorùbá and Hausa.

1. AG-News: originates from AG, which is a large collection of news articles. Zhang, Zhao, and LeCun (2015a) constructed the AG-News dataset from the AG collection and it is used as a benchmark dataset for text classification.

2. IMDB: consists of movie reviews with binary labels. It is a commonly used benchmark dataset used for text classification.

3. Yorùbá: The dataset was created from BBC Yorùbá news titles along with the noisy dataset (Hedderich et al., 2020).

4. Hausa: Similar to Yorùbá, the Hausa dataset and the corresponding noisy dataset were created by Hedderich et al. (2020) from VOA Hausa news titles and distant supervision using keywords.

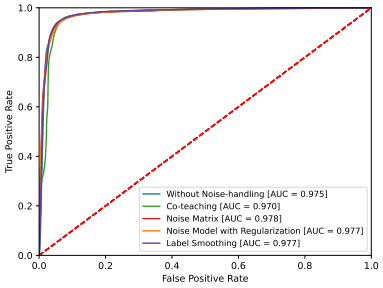
	Average Runtime (Hours)			
	AG-News	IMDB	Yorùbá	Hausa
CT	5	4.5*	0.1*	0.1*
NMat	2.5	8	0.1*	0.1*
NMwR	3	8	0.1*	0.1*
LS	2.5	8	0.1*	0.1*
WN	2.5	8	0.1*	0.1*

Table A.5: Average runtime (in hours) of each method. The Numbers with "*" indicates that the experiment was run on a Nvidia Tesla V100. Other experiments were run on a Nvidia GeForce GTX TITAN X.

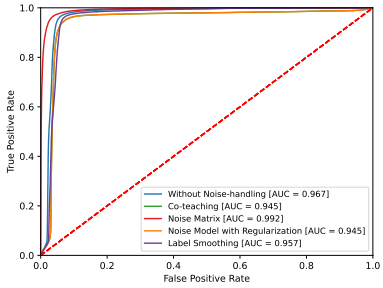
MODELS We use the official BERT-base model (Devlin et al., 2019) for text classification on AG-News and IMDB. It consists of an embedding layer, a 12-layer encoder, and a pooling layer. It contains 110M parameters in total. We use the multilingual version of BERT-base for text classification on Yorùbá and Hausa. It has the same architecture as the original BERT-base model. It also has 110M training parameters.

FINE-TUNING ON TEXT CLASSIFICATION TASK For the vanilla models (Wihtout Noise-handing and No Validation models in Chapter 3), we pass the final layer of the [CLS] token representation ($\mathbb{R}^{768}$) to a feed forward layer for prediction. Noise Matrix and Noise Matrix with Regularization append a noise matrix $N \in \mathbb{R}^{k \times k}$ after the model's prediction. For Noise Matrix we initialize the matrix with the ground truth information. Following (Jindal et al., 2019), when applying Noise Matrix with Regularization, we initialize the noise matrix using an identity matrix. The hyper-parameters for Noise Matrix with Regularization, Co-teaching and Label Smoothing are chosen so that the model performs the best on the noisy validation set.
In all settings, a batch size of 32 is used, and the learning rate is set to 2e-5. We train all models until the training loss converges. However, we report the score where the model performs the best on the validation set during training except for the No Validation baseline where we report the last-epoch performance.

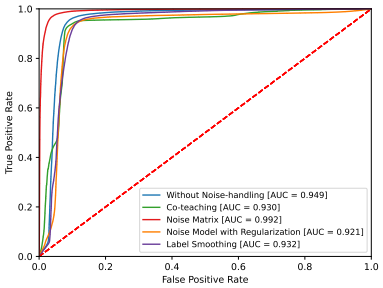
HARDWARE AND AVERAGE RUNTIME We use Nvidia Tesla V100 and Nvidia GeForce GTX TITAN X to accelerate training. The average runtime for each method and dataset is summarized in Table A.5.



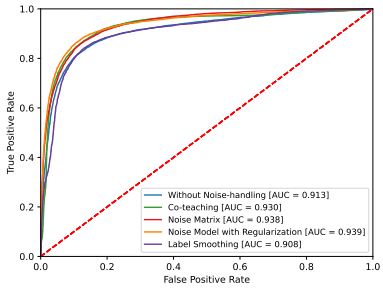
(a) AG-News - 60% uniform noise



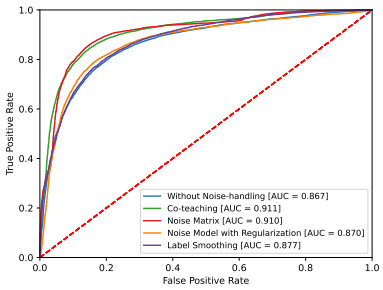
(b) AG-News - 40% single-flip noise



(c) AG-News - 45% single-flip noise



(d) IMDB - 40% single-flip noise



(e) IMDB - 45% single-flip noise

Figure A.3: The losses are recorded at the training step when early-stopping is triggered. Noise-handling methods do not make the losses of correct and incorrect labels more distinguishable.

B

A META-LEARNING BASED NOISE-HANDLING METHOD

B.1 DATASET DETAILS

We experiment with eight Natural Language Processing (NLP) datasets, including six English datasets and two datasets in low-resource languages. All datasets come with their ground truth annotations and as well as the weak labels.

B.1.1 *Datasets Selection Criteria*

The WRENCH (Zhang et al., 2021c) benchmark contains 23 NLP datasets. We choose representative datasets (like previous research in weak supervision) that **a)** overlap with previous works to enable direct comparisons. **b)** are diverse in terms of weak label quality, languages and tasks to approve the applicability of different baselines.

B.1.2 *English Datasets*

We experiment with four popular sequence classification datasets: AGNews, IMDB, Yelp and TREC.

1. **AGNews** (Zhang, Zhao, and LeCun, 2015b): originates from AG, which is a large collection of news articles. The news are categorized in four classes: "World", "Sports", "Business" and "Sci/Tech".

2. **IMDB** (Maas et al., 2011): consists of movie reviews with binary labels. It is a commonly used benchmark dataset for sentiment analysis.

3. **Yelp** (Zhang, Zhao, and LeCun, 2015b): obtained from the Yelp Dataset Challenge in 2015. Similar to IMDB, it is a sentiment analysis dataset.

4. **TREC** (Li and Roth, 2002): categorizes the questions in TREC-6 datasets into 6 categories: "Abbreviation", "Entity", "Description", "Human", "Location", "Numeric-value".

and with the two sequence labeling datasets: CoNLL-03 and OntoNotes 5.0.

1. **CoNLL-03** (Tjong Kim Sang and De Meulder, 2003) NER dataset with four named-entity categories.

2. **OntoNotes 5.0** (Pradhan et al., 2013): NER dataset with 18 named-entity categories.

All weak labels are obtained from the WRENCH benchmark[1] (Zhang et al., 2021c).

B.1.3 *Datasets in Low-Resource Languages*

Most datasets in the current WRENCH benchmarks are in English. Although weak supervision is desired in low-resource languages, it is understudied as finding annotators for them is more difficult. Hence, we include two low-resource languages, Yorùbá and Hausa, to cover this scenario. Often, learning with weak labels in low-resource languages is more challenging. First, the training set is often much smaller than English datasets. For example, Hausa has only about 2k training samples while AGNews have 96k. Second, the weak labels in low-resource languages can have lower quality as experts for weak source development are harder to find. A set of simple rules is often used for labeling (which is the case in Yorùbá and Hausa). Hence, weak supervision with low-resource languages is a combination of two challenges: training with *small* datasets which have *low-quality* labels.

Yorùbá and Hausa are text classification datasets obtained from (Hedderich et al., 2020).[2]

1. **Yorùbá**: consists of news headlines from BBC Yoruba which are categorized in seven classes: "Nigeria", "Africa", "World", "Entertainment", "Health", "Sport", "Politics".

2. **Hausa**: consists of news headlines from VOA Hausa which have the same seven classes as Yorùbá. However, only five classes are considered in the text classification task. "Entertainment" and "Sport" have been removed due to the lack of samples of these classes.

Hedderich et al. (2020) provided both the clean labels and weak labels on the two datasets. A gazetteer is created for each class for weak supervision. For example, a gazetteer containing names of agencies, organizations, states and cities in Nigeria is used to label the class "Nigeria".

B.1.4 *More Dataset Statistics*

We provide more details on the datasets we used in our experiments in Table B.1. In general, not all data can be covered by weak sources. No weak source is triggered for some training samples and they remain

1 https://github.com/JieyuZ2/wrench
2 https://github.com/uds-lsv/transfer-distant-transformer-african

Dataset	Task	# Class	$\lvert\mathcal{D}_w\rvert$	$\lvert\mathcal{D}_a\rvert$	Coverage	Conflict	$\lvert\mathcal{D}_v\rvert$	$\lvert\mathcal{D}_t\rvert$
AGNews	Topic	4	66,314	96,000	69.08%	14.17%	12,000	12,000
IMDB	Sentiment	2	17,515	20,000	87.58%	11.96%	2,500	2,500
Yelp	Sentiment	2	25,165	30,400	82.78%	18.29%	3,800	3,800
TREC	Question	6	4,723	4,965	95.13%	22.76%	500	500
Yoruba	Topic	7	1,340	1,340	100.00%	1.87%	189	379
Hausa	Topic	5	2,045	2,045	100.00%	1.90%	290	582
CoNLL03	NER	4	14,041	14,041	100.00%	4.05%	3,250	3,453
OntoNotes5.0	NER	18	115,812	115,812	100.00%	1.86%	5,000	22,897

Table B.1: Dataset statistics. $|\mathcal{D}_w|$: number of training samples with weak labels. $|\mathcal{D}_a|$: total number of training samples (weakly labeled + unlabeled). Coverage: fraction of samples that are weakly labeled, i.e., $\frac{|\mathcal{D}_w|}{|\mathcal{D}_a|}$. Conflict: samples that are labeled by at least two weak sources with contradicted weak labels. $|\mathcal{D}_v|$: number of validation samples. $|\mathcal{D}_t|$: number of test samples.

Hyperparameter	Search Range
Teacher Learning Rate	3e-6, 5e-6, 2e-5, 3e-5
Teacher Warm-Up Steps	500, 100, 2000, 3000
Confidence Filter Threshold	0.4, 0.5, 0.6, 0.7, 0.8, 0.95

Table B.2: Hyperparameter search.

unlabeled. The coverage of the datasets ranges from 69.08% to 100%. Note that for NER tasks, the coverage is always 100% since if no weak source is triggered for a token, we assign label "O" (i.e., non-entity) to it. On the other hand, some samples can be covered by two or more weak sources with contradicted weak labels. In this case, we have a conflict. The conflict ratio ranges from 1.86% to 22.76% in the datasets we tested.

	AGNews	IMDB	Yelp	TREC	Yorùbá	Hausa	CoNLL-03	OntoNotes 5.0
BERT Backbone	RoBERTa	RoBERTa	RoBERTa	RoBERTa	mBERT	mBERT	RoBERTa	RoBERTa
Batch Size	32	16	16	32	32	32	32	32
Maximum Sequence Length	128	256	256	64	64	128	64	64
Student Learning Rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Teacher Learning Rate	2e-5	2e-5	2e-5	2e-5	5e-6	2e-5	2e-5	2e-5
Teacher Warm-Up Steps	500	500	3000	500	1000	3000	2000	2000
Confidence Filter Threshold	0.7	0.7	0.5	0.5	0.7	0.4	0.8	0.5

Table B.3: Selected hyperparameters. mBERT: multilingual BERT.

B.2 IMPLEMENTATION DETAILS

MODELS. All baselines in Chapter 4, except the majority vote and the Snokerl model (Ratner et al., 2017) which work with label space only, use the official RoBERTa model[3] (Liu et al., 2019) from Hugging-

3 https://huggingface.co/roberta-base

face as the classification backbone for all English datasets, and the multilingual BERT[4] for datasets in African languages. We use the base version of the two models which contain roughly 120M and 110M parameters, respectively.

Dataset	Test	Validation
AGNews	89.92	89.90
IMDB	89.16	89.21
Yelp	95.00	94.79
TREC	94.80	94.42
Yorùbá	72.56	75.13
Hausa	59.11	62.34
CoNLL-03	88.41	87.86
OntoNotes	74.59	75.20

Table B.4: The average test and validation accuracy/F1 score (in %) of MSR over five trials.

MSR Configuration	AGNews (Acc)	IMDB (Acc)	Yelp (Acc)	TREC (Acc)	Yorùbá (Acc)	Hausa (Acc)	CoNLL-03 (F1)	OntoNotes (F1)
Student	**89.92**	**89.16**	**95.00**	**94.80**	**72.56**	59.11	**88.41**	**74.59**
Teacher	89.02	88.08	94.37	93.80	68.87	**60.14**	87.30	73.22
w/o Teacher Scheduler	89.68	87.68	93.78	93.60	70.71	55.32	87.82	72.48
w/o Confidence Filtering	89.87	89.04	94.76	93.60	71.50	55.15	88.07	74.11
w/o Both	89.55	87.68	93.33	93.40	70.50	55.32	87.82	72.08

Table B.5: Ablation studies. The numbers represent the test accuracy and F1 Score.

	AGNews	IMDB	Yelp	TREC	Yorùbá	Hausa	CoNLL-03	OntoNotes 5.0
Running time (hours)	2.5	1.6	0.5	1.2	0.5	0.7	1.1	3.0

Table B.6: Average runtime (in hours) for training a MSR model. One single Nvidia Tesla V100 GPU is used in each experiment to accelerate the computation.

FINE-TUNING ON CLASSIFICATION TASK. We fine-tune all layers using AdamW (Loshchilov and Hutter, 2019) as the optimizer. For sequence classification tasks, we pass the final layer of the [CLS] token representation ($\mathbb{R}^{768}$) to a feed-forward layer for prediction. For sequence labeling tasks, the final layers of all tokens ($\mathbb{R}^{768\times L}$, where L is the sentence length) are passed to a shared feed-forward layer to predict the class of each token in the sentence. We report the score where the model performs the best on the validation set during training.

4 https://huggingface.co/bert-base-multilingual-cased

HYPER-PARAMETERS OF MSR. We apply grid search on the warm-up steps for the teacher and the confidence threshold for the student network. Table B.2 shows our hyperparameter search configuration. We choose the final configurations of the hyperparameters according to the model's performance on the validation set. Table B.3 shows the best configurations of parameters we used to produce the results in Table 4.2.

EVALUATION METRICS. For model evaluation, we report accuracy for sequence classification tasks and F1 Score for sequence labeling tasks. In our implementation, we call the function `classification_report()` from the scikit-learn library[5] to compute the accuracy, and use the `Seqeval` class from Huggingface[6] to compute the F1 Score.

B.3 VALIDATION PERFORMANCE

The average test performance of MSR is reported in Table 4.2. We further report the corresponding validation performance in Table B.4.

B.4 ABLATION STUDIES

We report the detailed ablation results for each dataset in Table B.5.

B.5 HARDWARE AND AVERAGE RUNTIME.

We use Nvidia Tesla V100 to accelerate training. The average runtime for each method and dataset is summarized in Table B.6.

5 https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

6 https://github.com/huggingface/datasets/blob/master/metrics/seqeval/seqeval.py

C

REALISTIC FEATURE-DEPENDENT NOISE HANDLING

C.1 DATASETS

					Avg. over labeling functions (LFs)							
Dataset	**Task**	**#Classes**	**#LFs**	**%Ovr. Coverage**	**%Coverage**	**%Overlap**	**%Conflict**	**%Prec.**	**MV**	**#Train**	**#Dev**	**#Test**
AGNews	News Class.	4	9	69.08	10.34	5.05	2.43	81.66	81.23	96,000	12,000	12,000
IMDb	Movie Sentiment Class.	2	5	87.58	23.60	11.60	4.50	69.88	73.86	20,000	2,500	2,500
Yelp	Business Sentiment Class.	2	8	82.78	18.34	13.58	4.94	73.05	73.31	30,400	3,800	3,800
TREC	Question Class.	6	68	95.13	2.55	1.82	0.84	75.92	62.58	4,965	500	500
SemEval	Web Text Relation Class.	9	164	100.00	0.77	0.32	0.14	97.69	77.33	1,749	200	692
ChemProt	Chemical Relation Class.	10	26	85.62	5.93	4.40	3.95	46.65	55.12	12,861	1,607	1,607
CoNLL-03	English News NER	4	16	100	100	4.30	1.44	72.19	60.38	14,041	3250	3453
OntoNotes 5.0	Multi-Domain NER	18	17	100	100	1.55	0.54	54.84	58.92	115,812	5,000	22,897

Table C.1: Detailed data statistics. Note that 'Class.' is an abbreviation for classification. Coverage is the amount of samples a labeling function (LF) matches. For NER datasets, labeling functions return an entity or "O" thus coverage is always 100%. Overlap asks how many samples have at least 2 matching labeling functions. MV (majority vote) performance is given as F1-score for the NER datasets and as accuracy on the test set otherwise.

In the following, we give a more comprehensive description of the datasets used. A subset of the commonly used WRENCH (Zhang et al., 2021c) benchmark is used, covering various aspects such as task type, coverage and dataset size. There is a total of four classification, two relation extraction and two sequence labeling datasets. See Table C.1 for a detailed set of data statistics.

AGNEWS (Zhang, Zhao, and LeCun, 2015b) is a topic classification dataset. The task is to classify news articles into four topics, namely world, sports, business and Sci-Fi/technology. Each labeling function is composed of multiple keywords to search for. The number of keywords differs from a few up to dozens.

IMDB (Maas et al., 2011) is a dataset of movie reviews sampled from the IMDb website. The task is binary sentiment analysis. The labeling functions are composed of keyword searches and regular expressions.

YELP (Zhang, Zhao, and LeCun, 2015b) is another sentiment analysis dataset, containing crowd-sourced business reviews. The labeling functions are created using keywords and a lexicon-based sentiment analysis library.

TREC (Li and Roth, 2002) is a question classification dataset, i.e., it asks what type of response is expected. The labels are abbreviation,

Label	Labeling Function
POS	beautiful, handsome, talented
NEG	than this, than the film, than the movie
POS	.*(highly | do | would | definitely | certainly | strongly | i | we).*(recommend | nominate).*
POS	.*(high | timeless | priceless | HAS | great | real | instructive).*(value | quality | meaning | significance).*

Table C.2: Examples of two keyword based and two regular expression based rules for the IMDb dataset.

Label	Labeling Function
ABBREVIATION	(|^)(what | what)[^\w]* (\w+){0,1}(does | does)[^\w]* ([^\s]+)*(stand for)[^\w]*(|$)
DESCRIPTION	(|^)(explain | describe | how | how)[^\w]* (\w+){0,1}(can | can)[^\w]*(|$)
ENTITY	(|^)(which | what | what)[^\w]* ([^\s]+)*(organization | trust | company | company)[^\w]*(|$)
HUMAN	(|^)(who | who)[^\w]*(|$)
LOCATION	(|^)(which | what | where | where)[^\w]* ([^\s]+)*(situated | located | located)[^\w]*(|$)
NUMERIC	(|^)(by how | how | how)[^\w]* (\w+){0,1}(much | many | many)[^\w]*(|$)

Table C.3: Rules for the TREC dataset. For each label a representative labeling function is given.

description and abstract concepts, entities, human beings, locations or numeric values. The labeling functions are created using regular expressions and make a lot of use of question words such as "what", "where" or "who".

SEMEVAL (Hendrickx et al., 2010) is a relation classification dataset, using nine relation types. Examples for relation labels are cause-effect, entity-origin or message-topic. Labeling functions are created using entities within a regular expression.

CHEMPROT (Krallinger et al., 2017) is another relation classification dataset, focusing on chemical research literature. It contains ten different types of relations, for example chemical-protein relations such as “biological properties upregulator”. The labeling functions are created using rules.

CONLL-03 (Tjong Kim Sang and De Meulder, 2003) is a NER dataset, with labels for the entities "person", "location", "organization", and "miscellaneous". Labeling functions are built using previously trained keywords, regular expressions and NER models.

ONTONOTES 5.0 (Pradhan et al., 2013) is an another NER dataset, using more fine-grained entities as CoNLL-03. Here, a subset of the CoNLL weak labeling sources is combined with keyword and regular expression based weak labeling sources.

Label	Labeling Function
Cause-Effect(e1,e2)	SUBJ-O caused OBJ-O
Component-Whole(e1,e2)	SUBJ-O is a part of the OBJ-O
Content-Container(e1,e2)	SUBJ-O was contained in a large OBJ-O
Entity-Destination(e1,e2)	SUBJ-O into OBJ-O
Entity-Origin(e1,e2)	SUBJ-O emerged from the OBJ-O
Instrument-Agency(e2,e1)	SUBJ-O took the OBJ-O
Member-Collection(e2,e1)	SUBJ-O of different OBJ-O
Message-Topic(e1,e2)	SUBJ-O states that the OBJ-O
Product-Producer(e1,e2)	SUBJ-O created by the OBJ-TITLE

Table C.4: One labeling function for each label of the SemEval dataset. Here e1 and e2 are entities which are already available in the dataset.

Label	Labeling Function
PERSON	RegEx searching list one of 7559 first names, followed by an upper-cased word
LOCATION	List of 15205 places
ORGANIZATION	WTO, Starbucks, mcdonald, google, Baidu, IBM, Sony, Nikon
MISCELLANEOUS	List of countries, languages, events and facilities

Table C.5: For each label, one labeling function of the CoNLL-03 dataset is displayed.

C.2 LABELING FUNCTIONS

Weak labeling sources are often abstracted as labeling functions and vary in aspects such as coverage, precision, or overlap (Karamanolakis et al., 2021; Ratner et al., 2017). To showcase how the weak labeling process works, a selection of examples of labeling functions is presented. More specifically, we provide examples of rules for the two classification datasets IMDb (Table C.2) and TREC (Table C.3), the relation classification dataset SemEval (Table C.4) and the NER dataset CoNLL-03 (Table C.5).

C.3 OVERALL IMPLEMENTATION DETAILS

This section summarizes the overall implementation details of WSL approaches used in Chapter 5. Refer to Appendix C.4 for hyperparameter configurations of PEFT approaches. We use the PyTorch framework[1] to implement all approaches discussed in Chapter 5. Hugging Face (Wolf et al., 2020) is used for downloading and training the RoBERTa-base model. AdapterHub (Pfeiffer et al., 2020) is used for implementing parameter-efficient fine-tuning.

HYPERPARAMETERS We implemented five WSL methods: FT (Devlin et al., 2019), L2R (Ren et al., 2018), MLC (Zheng, Awadallah, and Dumais, 2021), BOND (Liang et al., 2020), and COSINE (Yu et al., 2021). We report the search ranges of the hyperparameters in Table C.6.

We do not search for batch size as we find it has minor effects on the final performance. Instead, a batch size of 32 is used across experiments. Also, RoBERTa-base (Liu et al., 2019) is used as the backbone PLM and AdamW (Loshchilov and Hutter, 2019) is the optimizer used across all methods.

COMPUTING INFRASTRUCTURE AND TRAINING COST We use Nvidia V100-32 GPUs for training deep learning models. All WSL approaches studied in Chapter 5 can fit into one single GPU. We report the training time of the WSL methods in Table C.7.

C.4 TRAINING WITH CLEAN SAMPLES

C.4.1 *Methods and implementation details*

In Section 5.6, we apply four (parameter-efficient) fine-tuning approaches to train models on clean validation sets. Since we do not have extra data for model selection, we choose a fixed set of hyperparameters for all datasets. In the following we briefly introduce the

1 https://pytorch.org/

Hyperparameter	**Search Range**
FT	
Learning rate	2e-5, 3e-5, 5e-5
Warm-up steps	50, 100, 200
L2R (Ren et al., 2018)	
Learning rate	2e-5, 3e-5, 5e-5
Meta-learning rate	1e-4, 2e-5, 1e-5
MLC (Zheng, Awadallah, and Dumais, 2021)	
Learning rate	2e-5, 3e-5, 5e-5
Meta-learning rate	1e-4, 2e-5, 1e-5
hdim	512, 768
BOND	(Liang et al., 2020)
Learning rate	2e-5, 3e-5, 5e-5
T_1	5000
T_2	5000
T_3	50, 100, 300, 500
Confidence threshold	0.1, 0.3, 0.5, 0.7, 0.8, 0.9
COSINE (Yu et al., 2021)	
Learning rate	2e-5, 3e-5, 5e-5
T_1	5000
T_2	5000
T_3	50, 100, 300, 500
Distance measure	cosine
Regularization factor	0.05, 0.1, 0.2
Confidence threshold	0.1, 0.3, 0.5, 0.7, 0.8, 0.9

Table C.6: The search range of the hyperparameters of the five WSL approaches considered in Chapter 5. For BOND and COSINE, we set T_1 and T_2 to constant values, because we stop training once early-stopping is triggered.

	AGNews	IMDb	Yelp	TREC	SemEval	ChemProt	CoNLL-03	OntoNotes 5.0
FT	0.2	0.2	0.2	0.1	0.1	0.2	0.2	0.5
L2R	2.0	1.2	1.5	0.3	0.3	0.4	0.9	1.2
MLC	1.2	0.8	1.2	0.3	0.2	0.5	1.2	1.0
BOND	0.5	0.2	0.5	0.1	0.1	0.2	0.4	1.1
COSINE	0.6	0.2	0.6	0.2	0.2	0.3	0.5	1.5

Table C.7: Running time in hours of each WSL method when trained on a weakly labeled training set. Since we also track the validation and test performance during training, the training time reported here actually overestimates the training time required for each method.

fine-tuning approaches, together with their hyperparameter configurations.

- Vanilla fine-tuning (Devlin et al., 2019; Liu et al., 2019) is the standard fine-tuning approaches for pre-trained language models. It works by adding a randomly initialized classifier on top of the pre-trained model and training it together with all other model parameters. We use a fixed learning rate of $2e^{-5}$ in all experiments.
- Adapter-based fine-tuning (Houlsby et al., 2019) adds additional feed-forward layers called adapters to each layer of the pre-trained language model. During fine-tuning, we only update the weights of these adapter layers and keep all other parameters *frozen* at their pre-trained values. We use a fixed learning rate of $2e^{-5}$ in all experiments. The reduction factor is set to 16.
- BitFit (Zaken, Goldberg, and Ravfogel, 2022) updates only the bias parameters of every layer and keeps all other weights frozen. Despite its simplicity it has been demonstrated to achieve similar results to adapter-based fine-tuning. We use a fixed learning rate of $1e^{-4}$ in all experiments.
- LoRA (Hu et al., 2022) is a recently proposed adapter-based fine-tuning method which uses a low-rank bottleneck architecture in each of the newly added feed-forward networks. The motivation here is to perform a low rank update to the model during fine-tuning. We use a fixed learning rate of $2e^{-5}$ in all experiments. The α value used in LoRa is fixed to 16.

In all experiments, the batch size used in all fine-tuning approaches is 32. The optimizer is AdamW (Loshchilov and Hutter, 2019).

C.4.2 *Training on the full validation sets*

In addition to training sets, the WRENCH (Zhang et al., 2021c) benchmark provides a validation set for each of its tasks. The validation sets are cleanly annotated and typically range in size from 5% to

25% of the weakly annotated training sets. Although such validation size is reasonable for fully supervised learning, we suspect that it is exorbitant in the sense that it provides a significantly better training signal for models than the weakly annotated training set. Thus we compare the performance of recent WSL approaches that access both the training and validation sets with a model that is directly fine-tuned on the validation set. The following WSL methods are included in this experiment: L2R (Ren et al., 2018), MetaWN (Shu et al., 2019), BOND (Liang et al., 2020), Denoise (Ren et al., 2020), MLC (Zheng, Awadallah, and Dumais, 2021), and COSINE (Yu et al., 2021). Following prior work, we select the best set of hyperparameters via the validation set when applying the WSL methods. Also, early-stopping based on the validation performance is applied. In contrast, the direct fine-tuning baseline uses a fixed set of hyperparameters across all datasets, and no early-stopping is applied (same configuration as in Appendix C.4.1). We train this baseline for 6000 steps. In all cases, the training losses converged much earlier than 6000 steps, but we deliberately kept training for longer to show that the good performance achieved by this baseline is not due to any fine-grained configurations. As shown in Figure 5.1, this simple baseline outperforms all the WSL methods in all but one case.

C.4.3 *Extended comparison of training on clean data and validation for WSL approaches*

In Section 5.6, standard fine-tuning (FT) and multiple parameter-efficient fine-tuning (PEFT) are compared with the competitive WSL method COSINE. In this section, we provide additional plots which show the same comparison with the other WSL methods examined, namely L2R, MLC, and BOND. We report average performance (Acc. and F1 in %) difference between (parameter-efficient) fine-tuning methods and the specific WSL method for varying number of clean samples. The overall tendency is consistent with the results in Section 5.6: WSL methods perform well on a small amount of clean labeled data but PEFT outperforms WSL methods with an increasing amount of clean labeled data.

C.5 ADDITIONAL BASELINES THAT COMBINE WEAK AND CLEAN DATA DURING TRAINING

Besides CFT we also explored two simple baselines that combine both the cleanly and weakly annotated data in training:

1. $\mathbf{WC_{mix}}$: it mixes the clean data into the weakly labeled training set. We then fine-tune a PLM on this combined dataset.

2. **WC$_{\text{batch}}$**: in each batch, we mix the weakly and cleanly labeled data at a ratio of 50:50. This makes sure that the model can access clean samples in each batch.

We compared these two baselines with CFT, the results are shown in Figure C.3. It can be seen that when the same amount of data is accessed, CFT outperforms the two baselines in most cases, sometimes by a large margin.

C.6 ADDITIONAL PLOTS ON CFT WITH DIFFERENT NUMBERS OF CLEAN SAMPLES

We show further plots of experiments in Section 5.7 with different numbers of clean samples in Figure C.4. More specifically, it shows the results for selecting $N \in \{10, 20, 30, 40\}$ clean samples per class from the clean validation set for classification and $N \in \{100, 200, 300, 400\}$ for NER tasks. These results corroborate the analysis presented in Section 5.7.

C.7 CFT WITH DIFFERENT PLMS AND AGREEMENT RATIOS

We provide additional plots of the experiments mentioned in Section 5.8 on more datasets. Figure C.5 shows the performance of CFT using different PLMs during training and Figure C.6 shows the performance when the number of clean samples and the agreement ratio is varied.

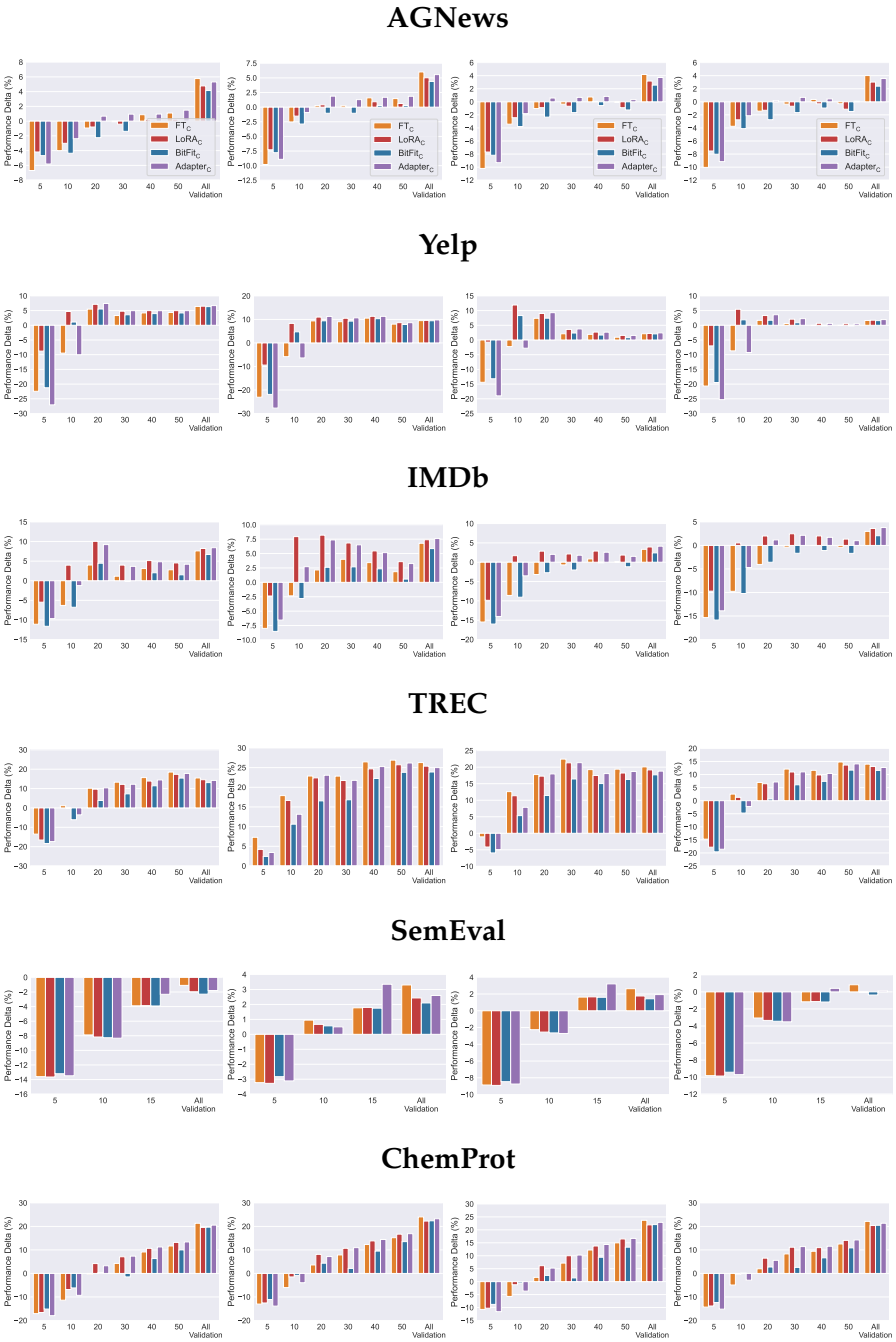


Figure C.1: Performance comparison of parameter-efficient fine-tuning methods (FT, LoRA, BitFit, and Adapter) with weakly supervised learning approaches (L2R, MLC, BOND, and COSINE). Evaluated on AGNews, Yelp, IMDb, TREC, SemEval, and ChemProt using varying amounts of clean data. The subscript "C" (e.g., FT_C) indicates that the fine-tuning methods are applied to clean data.

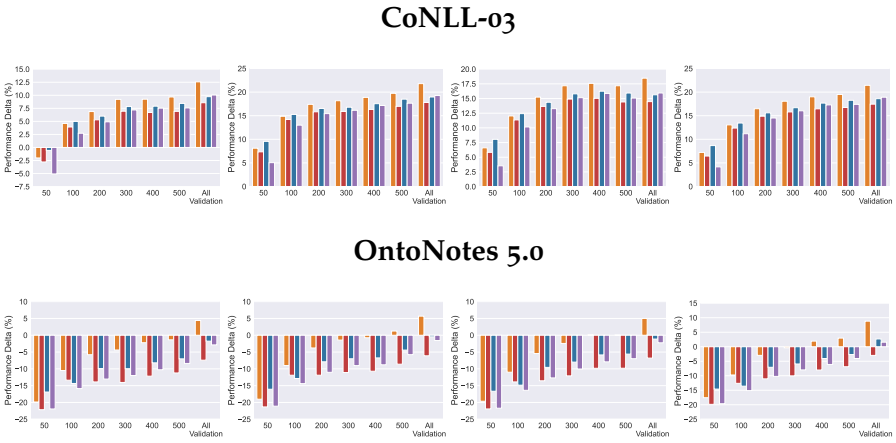
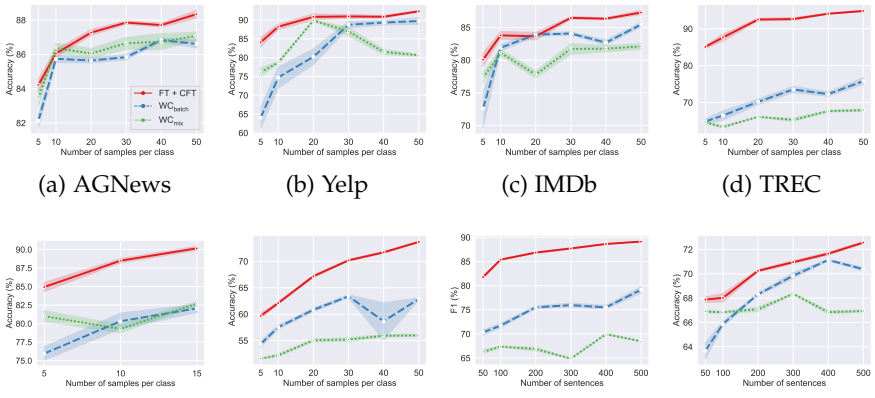


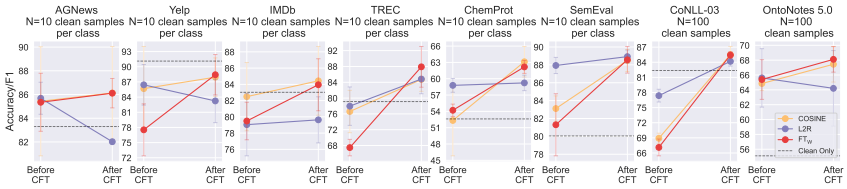
Figure C.2: Performance comparison of parameter-efficient fine-tuning methods (FT, LoRA, BitFit, and Adapter) with weakly supervised learning approaches (L2R, MLC, BOND, and COSINE). Evaluated on CoNLL-03 and OntoNotes 5.0 using varying amounts of clean data. The subscript "C" (e.g., FT_C) indicates that the fine-tuning methods are applied to clean data.



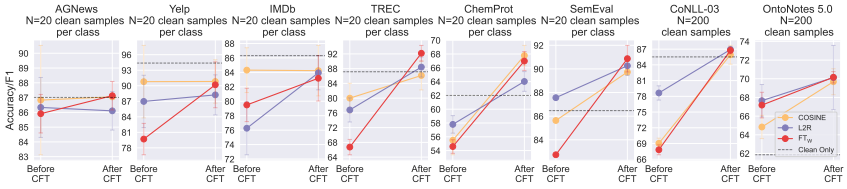
(a) AGNews (b) Yelp (c) IMDb (d) TREC

(e) SemEval (f) ChemProt (g) CoNLL-03 (h) OntoNotes 5.0

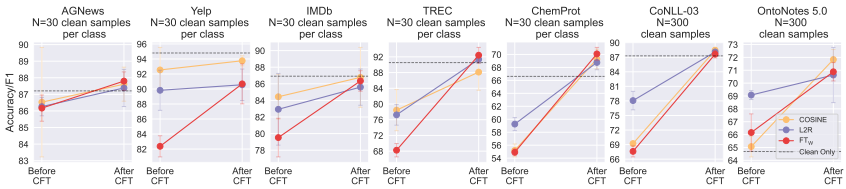
Figure C.3: Performance vs. number of clean samples. In most cases, CFT outperforms the other two baselines, WC_{batch} and WC_{mix}, by a considerable margin.



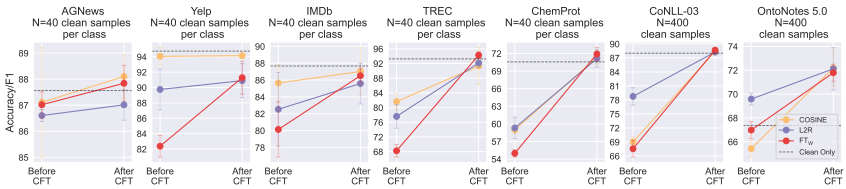
(a) $N = 10$ samples per class ($N = 100$ sentences on NER)



(b) $N = 20$ samples per class ($N = 200$ sentences on NER)

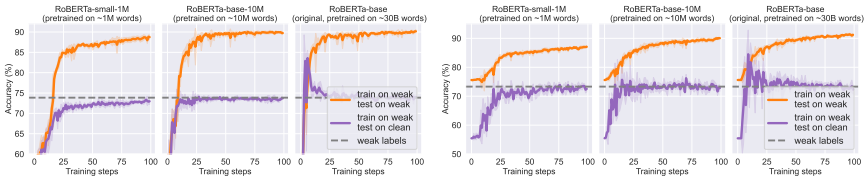


(c) $N = 30$ samples per class ($N = 300$ sentences on NER)



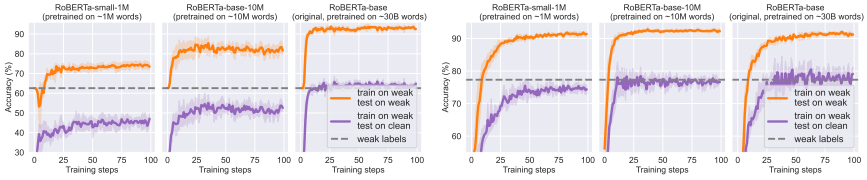
(d) $N = 40$ samples per class ($N = 400$ sentences on NER)

Figure C.4: Performance difference before and after applying CFT to WSL methods. For text classification and relation extraction tasks, we subsample $N \in \{5, 10, 20, 30, 40, 50\}$ examples from the validation set. For NER, we subsample $N \in \{50, 100, 200, 300, 400, 500\}$. On SemEval, the original validation set is small, and sampling more than 20 samples per class is not possible. The figure shows that the performance gap between the simple baseline FT_W and COSINE/L2R becomes much smaller after CFT, suggesting that we may not require sophisticated WSL methods to achieve good generalization.



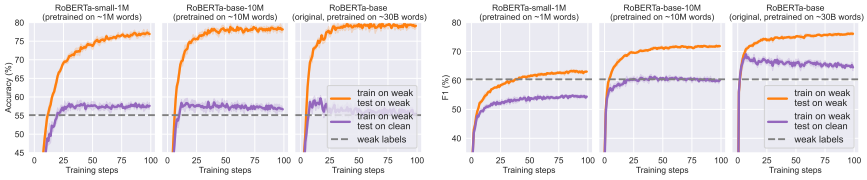
(a) IMDb

(b) Yelp



(c) TREC

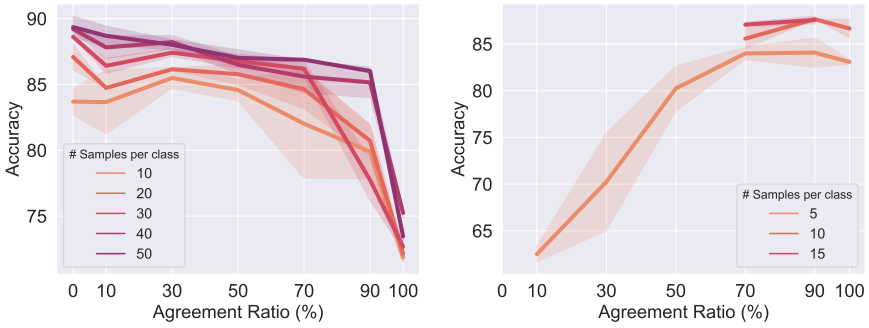
(d) SemEval



(e) ChemProt

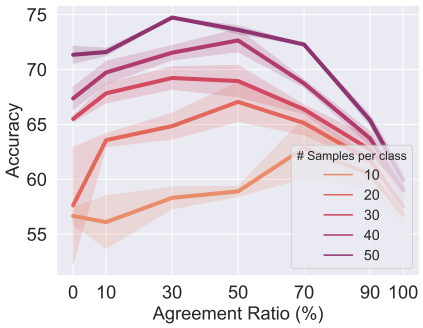
(f) CoNLL-03

Figure C.5: **Performance curves of different PLMs during training**. PLMs are trained on weak labels and evaluated on both clean and weakly labeled test sets. Pre-training on larger corpora improves performance on the clean distribution.

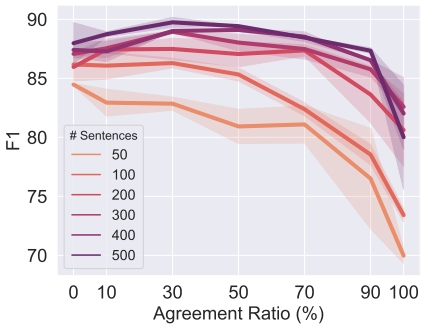


(a) IMDb

(b) SemEval*



(c) ChemProt



(d) CoNLL-03

Figure C.6: **Model performance varying the number of clean samples N and agreement ratio α.** Large values of α generally cause a substantial performance drop. *: Certain combinations of α and N are not feasible because the validation set lacks samples with clean and weak labels that coincide or differ.

D

FEATURE-DEPENDENT NOISE IN MACHINE TRANSLATION

D.1 MODEL PERFORMANCE WITH VARYING TRAINING SAMPLE SIZES

In Figure D.1 and Figure D.2, we present the performance for instruction-tuned baselines and our models on different evaluation directions. For most directions, using only 32 training samples can achieve competitive performance and beat all three instruction-tuned baselines. There are several exceptional cases, including en→zh and en→ja, in which the COMET score of SFT with a limited number of samples (32 or 64) is worse than 1-shot in-context learning.

While we primarily report the results with Llama-2 7B in our experiments, we hypothesize that state-of-the-art LLMs are largely homogeneous in terms of language distribution and inherent translation capability making our findings applicable to other LLMs. To support this hypothesis, we conduct fine-tuning experiments with Mistral 7B and Llama-2 13B using varying data sizes: 32, 1024, and 70K. As shown in Figure D.3, the general trend is quite similar to the Llama-2 7B case: fine-tuning with 32 examples results in competitive performance, matching or surpassing general-purpose instruction-tuned models. Furthermore, increasing the number of training examples leads to diminishing returns.

D.2 MODEL PERFORMANCE WITH VARYING TRAINING DIRECTIONS

Figure D.4 shows normalized BLEU scores for different combinations of train and test translation directions. Similar to the COMET scores in Figure 6.2, we observe that when training the model on a single direction, its translation ability across other non-targeted directions is also elicited to a certain degree. It is worth noting that when the training direction is X→en, the performance on directions en→X is significantly worse than training on all directions.

D.3 COMBINED EFFECT OF TRAINING SIZE AND DIRECTION

Figure D.7 illustrates the model performance across varying training sizes and translation directions, evaluated on en→cs, de, zh. Similarly, Figure D.8 presents the results on en→cs, de, zh, and en→hr. Consistently across all plots, we observe a positive impact on performance

with an increasing number of training directions, particularly with smaller training sizes.

D.4 MODEL PERFORMANCE WITH UNSEEN LANGUAGES

In Figure D.5, we find similar patterns as the COMET score, where fine-tuning on unseen languages can elicit the model's ability to translate from and to all seen languages. However, the translation performance on unseen languages themselves remains subpar, suggesting that SFT primarily reveals the knowledge LLMs have possessed during pre-training.

D.5 MODEL PERFORMANCE WITH NOISY DATA

Figure D.6 shows the BLEU score of different translation directions with two noise types. We can find that models are more sensitive to word-level noise than sentence-level noise. Also, the performance degradation is more noticeable when injecting noise into the source translation side. In comparison to the results of size 1024, using 32 training examples still achieves comparable or even better performance in the noisy condition.

D.6 TECHNICAL DETAILS

D.6.1 *Datasets*

Our parallel data is derived from the development and test sets of WMT17 through WMT22. Detailed dataset statistics are available in table D.1. For most experiments, we use the test sets from WMT17 to WMT20 for training. The test set from WMT22 is used specifically for testing. An exception is noted in Section section 6.3.4, where models are trained using the en↔ha and en↔is language pairs from WMT21's development set. Subsequently, these models are evaluated using the corresponding test sets from WMT21.

D.6.2 *Translation instructions*

The collection of translation instruction templates used in Chapter 6 can be found in table D.2.

D.6.3 *Evaluation packages*

To obtain COMET scores, we use `Unbabel/wmt22-comet-da`[1] and for BLEU scores, we use sacreBLEU[2] (Post, 2018). The signature from the sacreBLEU package is `nrefs:1, case:mixed, eff:no, tok:13a, smooth:exp, version:2.0.0` for all language pairs, except for tokenization for `en`→`zh` and `en`→`jp`, where we use `tok:zh` and `tok:jp-mecab`, respectively.

Direction	Training					Validation*	Test
	WMT17	WMT18	WMT19	WMT20	WMT21dev	WMT21	WMT22
en-cs	3005	2983	1997	1418	0	1002	2037
en-de	3004	2998	1997	1418	0	1002	2037
en-hr	0	0	0	0	0	0	1671
en-ja	0	0	0	1000	0	0	2037
en-ru	3001	3000	1997	2002	0	1002	2037
en-zh	2001	3981	1997	1418	0	1002	2037
cs-en	3005	2983	0	664	0	1000	1448
de-en	3004	2998	2000	785	0	1000	1984
ja-en	0	0	0	993	0	1005	2008
ru-en	3001	3000	2000	991	0	1000	2016
zh-en	2001	3981	2000	2000	0	1948	1875
en-ha	0	0	0	0	2000	1000	0
ha-en	0	0	0	0	2000	997	0
en-is	0	0	0	0	2004	1000	0
is-en	0	0	0	0	2004	1000	0
de-fr	0	0	1701	1619	0	⊗	1984
fr-de	0	0	1701	1619	0	⊗	2006

Table D.1: Data statistics. *Generally, WMT21 test is used for validation purposes; exceptions are en↔ha and en↔is, which are used for testing. ⊗Although WMT21 includes data for de↔fr, these language pairs are excluded from experiments.

D.6.4 *Hardware specifications and runtime*

Our experiments are conducted on a computing node with either 8 NVIDIA A100-40GB GPUs or 8 H100-80GB GPUs. DeepSpeed[3] with zero-stage 1 and mixed precision bfloat16 is used for performing SFT. Given the limited dataset size, typically fewer than 1024 samples, each SFT experiment can be completed within a mere 15 minutes using four H100 GPUs. However, given the necessity to evaluate the models across more than ten translation directions, the evaluation process

1 https://github.com/Unbabel/COMET
2 https://github.com/mjpost/sacrebleu
3 https://github.com/microsoft/DeepSpeed

may require up to four hours when performed on a single A100-40GB GPU.

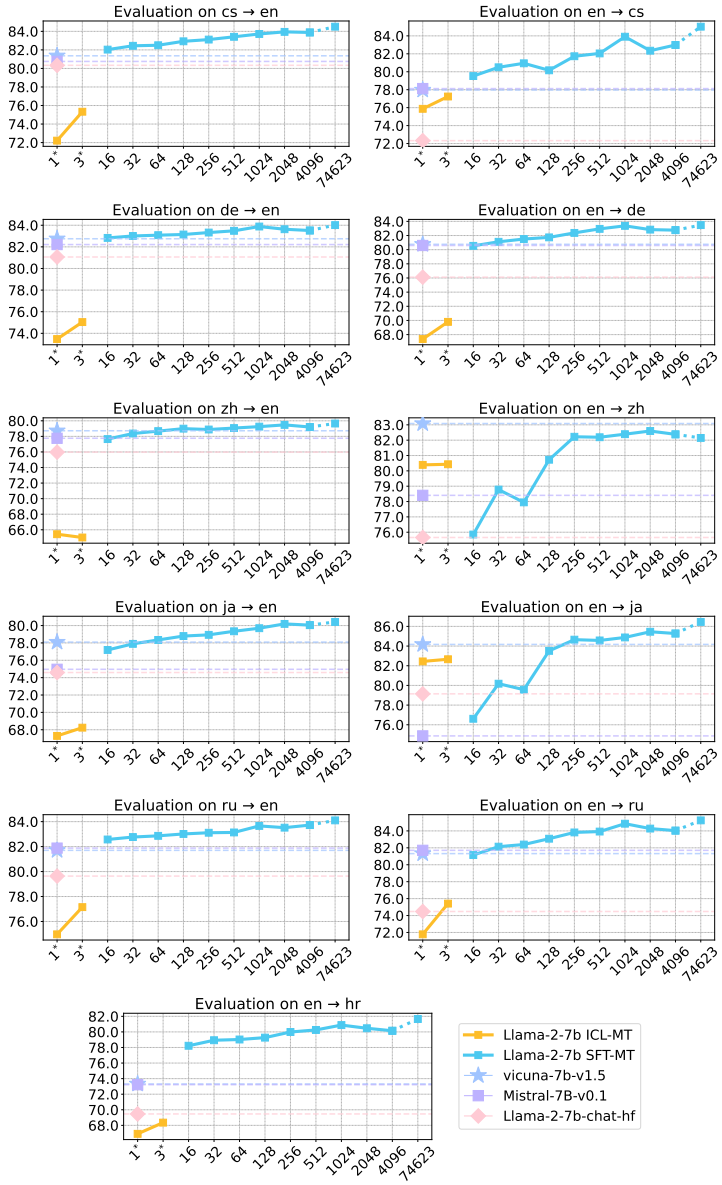


Figure D.1: COMET scores between instruction-tuned baselines and our models at different training data sizes, evaluated on individual translation directions. ICL is used for training sizes at or below 3, indicated with "*"; otherwise, we perform SFT. With only 32 examples for SFT, Llama-2 outperforms general-purpose, instruction-tuned baselines. Base.: instruction-tuned baseline models.

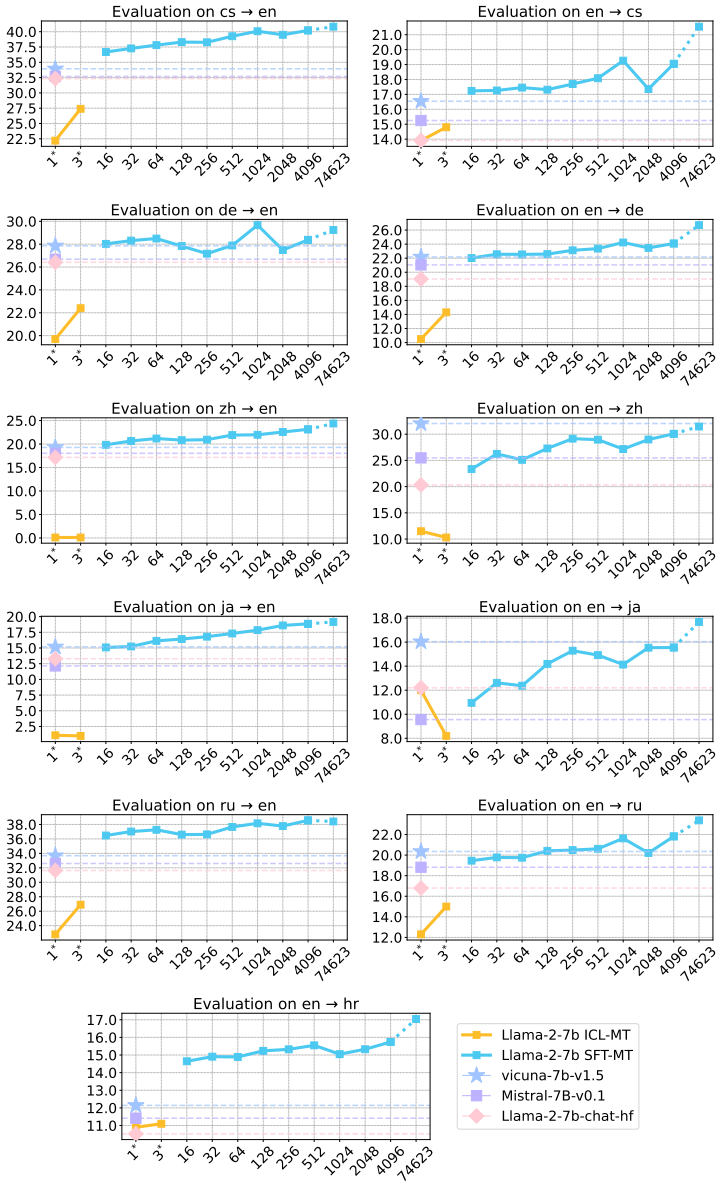


Figure D.2: BLEU scores between instruction-tuned baselines and our models at different training data sizes, evaluated on individual translation directions. ICL is used for training sizes at or below 3, indicated with "*"; otherwise, we perform SFT. With only 32 examples for SFT, Llama-2 outperforms general-purpose, instruction-tuned baselines. Base.: instruction-tuned baseline models.

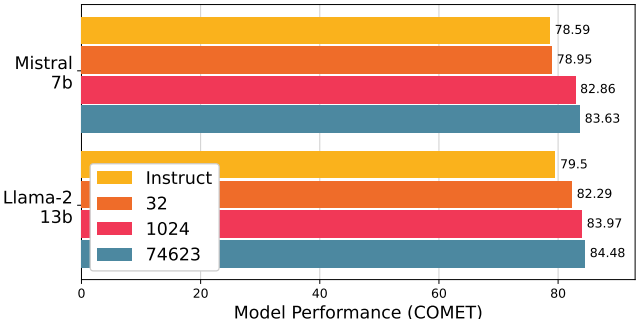


Figure D.3: Performance comparison between instruction-tuned baselines and fine-tuned models with different training data sizes. "Instruct" refers to the instruction-tuned baselines, specifically Mistral-7B-Instruct-v0.1 and Llama-2-13b-chat. "32/1024/74623" represents models fine-tuned on 32, 1024, and 74623 examples, using pre-trained only models: Mistral-7B-v0.1 and Llama-2-13b.

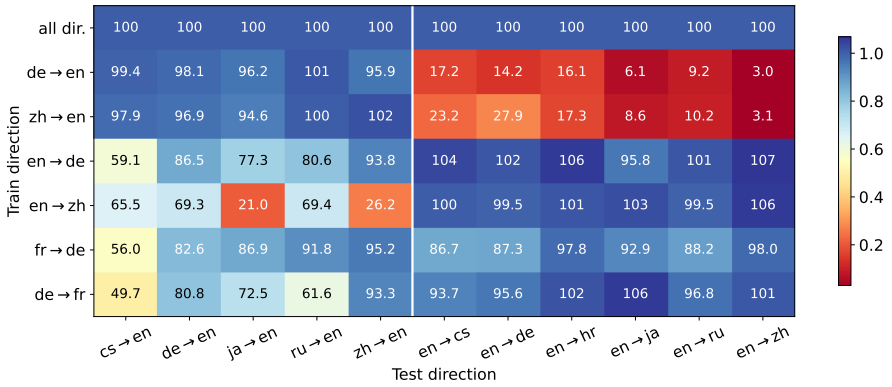


Figure D.4: Model performance (%) in BLEU score resulted from varying combinations of train and test translation directions. The scores are normalized according to Llama-2 fine-tuned on all 10 training directions.

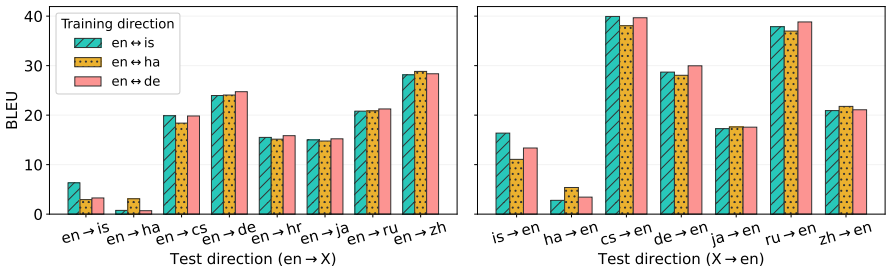


Figure D.5: Model performance evaluated across 15 translation directions. While models trained on *unseen* languages (en↔is, en↔ha) exhibit moderate improvements in translating these languages, they demonstrate accurate translations from and to *seen* languages.

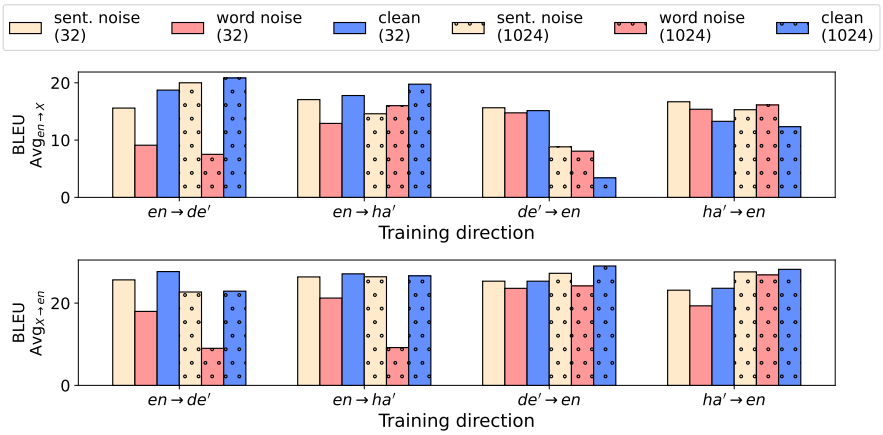


Figure D.6: Model performance in BLEU score varying training sizes, directions, and noise types. Top (Bottom): score averaged across all en→X (X→en) test directions. Training sizes considered are 32 and 1024.

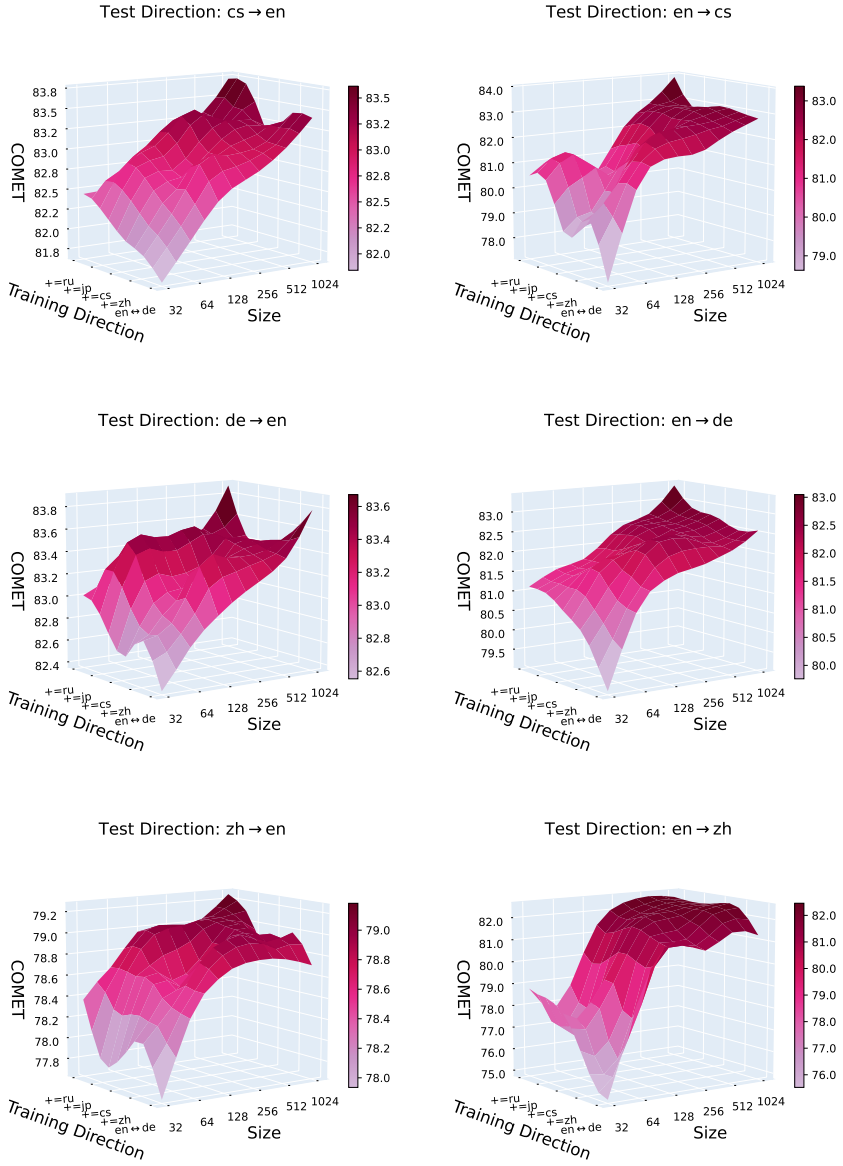


Figure D.7: Model performance (in COMET) on individual directions for models trained with varying data sizes and directions. Both factors positively impact performance. +=: training directions added on top of previous directions; two directions (from and to English) at a time. For example, "+=ru" covers 10 directions: en ↔ {de, zh, cs, jp, ru}.

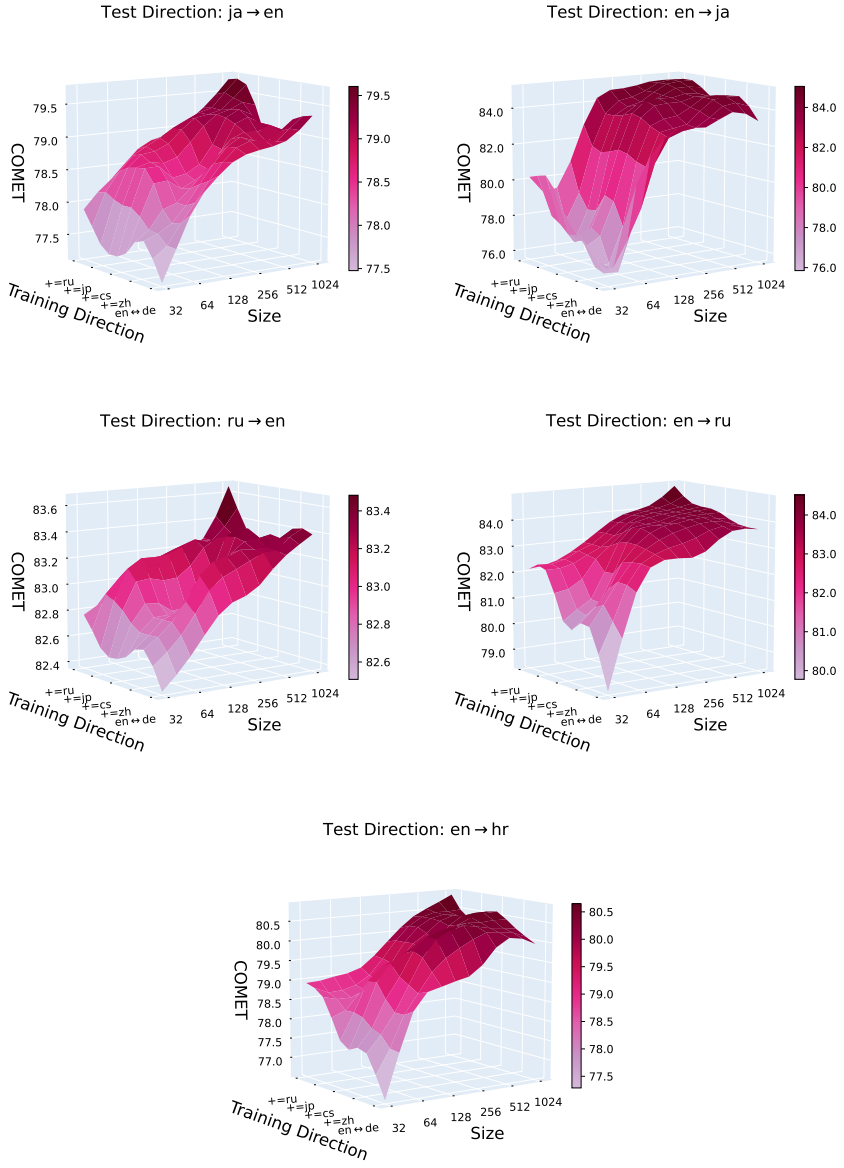


Figure D.8: Model performance (in COMET) on individual directions for models trained with varying data sizes and directions. Both factors positively impact performance. +=: training directions added on top of previous directions; two directions (from and to English) at a time. For example, "+=ru" covers 10 directions: en ↔ {de, zh, cs, jp, ru}.

Instruction pool
Please provide the [TGT] translation for the following text
Convert the subsequent sentences from [SRC] into [TGT]:
Render the listed sentences in [TGT] from their original [SRC] form:
Transform the upcoming sentences from [SRC] language to [TGT] language:
Translate the given text from [SRC] to [TGT]:
Turn the following sentences from their [SRC] version to the [TGT] version:
Adapt the upcoming text from [SRC] to [TGT]:
Transpose the next sentences from the [SRC] format to the [TGT] format.
Reinterpret the ensuing text from [SRC] to [TGT] language.
Modify the forthcoming sentences, converting them from [SRC] to [TGT].
What is the meaning of these sentences when translated to [TGT]?
In the context of [TGT], what do the upcoming text signify? The text is:
How would you express the meaning of the following sentences in [TGT]?
What is the significance of the mentioned sentences in [TGT]?
In [TGT], what do the following text convey?
When translated to [TGT], what message do these sentences carry?
What is the intended meaning of the ensuing sentences in [TGT]?
How should the following sentences be comprehended in [TGT]?
In terms of [TGT], what do the next sentences imply?
Kindly furnish the [TGT] translation of the subsequent sentences.
Could you supply the [TGT] translation for the upcoming sentences?
Please offer the [TGT] rendition for the following statements.
I'd appreciate it if you could present the [TGT] translation for the following text:
Can you deliver the [TGT] translation for the mentioned sentences?
Please share the [TGT] version of the given sentences.
It would be helpful if you could provide the [TGT] translation of the ensuing sentences.
Kindly submit the [TGT] interpretation for the next sentences.
Please make available the [TGT] translation for the listed sentences.
Can you reveal the [TGT] translation of the forthcoming sentences?
Translate from [SRC] to [TGT]:

Table D.2: A collection of 31 translation prompts. Each instruction is randomly selected to form a training sample. At inference time, the first instruction is always selected. The placeholders [SRC] and [TGT] represent the source and target languages, respectively, and will be replaced with the appropriate languages depending on the specific example at hand.

E

LEVERAGE IMPERFECT DATA IN MACHINE TRANSLATION

E.1 INCORPORATING MULTIPLE PREFERENCES WITH DISTANCE INFORMATION

In Section 7.3.2, we demonstrated how the distance information of two preferences can be integrated into preference modeling, as illustrated in Equation 7.9. A similar analysis can be done for the Plackett-Luce ranking model to incorporate distance metrics across multiple preferences. Specifically, we model the probability of a particular ordering $X_1, \cdots, X_L$ as follows:

$$P(X_1 \geq X_2 \cdots \geq X_L) = \prod_{i=1}^{L-1} P_i(X_i > X_j, \forall j > i)$$

For each distribution P_i, let $X_j = s_j + \varepsilon_j$ for $j \geq i$, with $\varepsilon_j \sim$ standard Gumbel and independent so that (following Train (2003), Section 3)

$$P_i(X_i > X_j, \forall j > i) = \frac{e^{s_i}}{\sum_{j \geq i} e^{s_j}}$$

This ranking can be interpreted as a sequence of $L-1$ independent choices: choose the first item, then choose the second among the remaining alternatives, etc. (Maystre and Grossglauser, 2015). It is usually assumed that each independent choice is made by the same judge whose underlying preferences do not change. If we assume $s_j = \log \pi_\theta(x, y^j)$ for this judge then Equation 7.4 results.

Suppose instead that, rather than a single judge, a succession of $L-1$ different judges each make one of the sequence of independent choices. The distributions P_i should change to reflect the changing preferences of the judges. In particular, if we introduce the preference distances d_i^j for the i^{th} judge, then we obtain Equation 7.5 if for each P_i the location parameters are set to $s_j = d_i^j \log \pi_\theta(x, y^j)$ for $j \geq i$. We find that this modified version of the Placket-Luce model can work well in practice although we note that these modifications may violate Luce's Choice Axiom (Hamilton, Tawn, and Firth, 2023; Luce, 1959).

Consider the case of $L = 3$. The Choice Axiom requires the odds of choosing X_2 over X_3 are independent of the presence of X_1 as an option, i.e. that the odds should not depend on whether this is a choice for the first or the second position

$$\frac{P_1(X_2 > X_j, j = 1, 3)}{P_1(X_3 > X_j, j = 1, 2)} = \frac{P_2(X_2 > X_3)}{P_2(X_3 > X_2)}$$

With the location parameters from above, the Choice Axiom requires

$$\frac{\pi_\theta(x, y^2)^{d_1^2}}{\pi_\theta(x, y^3)^{d_1^3}} = \frac{\pi_\theta(x, y^2)^{d_2^2}}{\pi_\theta(x, y^3)^{d_2^3}}$$

or that $\pi_\theta(x, y^2)^{(d_1^2 - d_2^2)} = \pi_\theta(x, y^3)^{(d_1^3 - d_2^3)}$. This holds for the default setting, $d_i^j = 1$, leading to Equation 7.4, but appears not to hold in general.

We find that the ground truth preference values can be introduced as preference distances in the binary comparison case, but that doing so in the more general case, while useful, may not satisfy the Axiom of Choice.

E.2 MORE DETAILS ON MAPLE

E.2.1 *Data Construction*

The source sentences in the training data of MAPLE are sampled from the test sets of WMT20 and WMT21. As mentioned in Section 7.4, four of the five translations are produced by VicunaMT. Considering that VicunaMT is already a strong MT system, often providing accurate translations free of mistakes, randomly selecting source sentences from WMT data could predominantly yield translations that are trivial for VicunaMT to translate, resulting in the collection of many uninformative samples with high human preference scores. To mitigate this, we prioritize source sentences that present difficulties for VicunaMT. Specifically, we use reference translations as a proxy to assess the quality of the model translations through COMET scores. We give priority to samples where the beam search output falls within a COMET score range of [75,85] and where there is a significant standard deviation in COMET scores among the four translations. Following these criteria, we select 1.1K samples for each translation direction. For the development set in MAPLE, we use monolingual data from News Crawl 2022. The sampling and selection process are the same as that of the training set, except that we do not have reference translations, instead, we use a strong commercial MT system to generate pseudo "reference" translations.

E.2.2 *Scoring Rubric*

The annotators are asked to judge the translation on a scale of 1 to 6, following the guidelines outlined in the following scoring rubric. They can assign scores in increments of 0.2, allowing for more detailed assessments, such as 1.2, 1.4, and so on.

- **Score it a 1** when the translation has nothing to do with the source; or when the translation has many unknown words; or when the translation looks like word salad.
- **Score it a 2** when you can understand why some of the words in the translation are there, but when the meaning of the source sentence is lost.
- **Score it a 3** when you understand why all or almost all the words in the translation are there and when some of the meaning of the source sentence are adequately transferred into the target language, but when the main meaning of the source sentence is lost.
- **Score it a 4** when the meaning of the source sentence is generally preserved, but when the translation is mechanical and possibly has vocabulary, grammatical, or date / numbering errors.
- **Score it a 5** when the meaning of the source sentence is fully preserved and the translation has no grammatical errors, but when the translation does not sound like the translation a native target language speaker would produce given the style and register of the source sentence.
- **Score it a 6** when the translation is perfect in every sense of the word – something a professional translator/interpreter would come up with when she understands well the context in which the source sentence was produced.

Figure E.1: User interface of translation assessment.

E.2.3 *Annotation UI*

The UI shows the different translations in a blind and randomized order. All translations are scored simultaneously. A screenshot of the UI is shown in Figure E.1.

Source	Other MPs criticised Twitter for allowing the tweets to remain visible.
Reference translation	其他议员也批评 Twitter 未能及时删贴。 *(Other MPs have also criticized Twitter for failing to promptly delete tweets in time.)*
Best of five translation	其他议员批评了推特允许这些推文仍然可见。 *(Other MPs have criticized Twitter for allowing these tweets to remain visible.)*
Source	When he refused, the officials tipped his cart over, destroying all the eggs, the boy alleged.
Reference translation	男孩说，他拒绝交出100卢比后，那些官员就把他的小车掀翻，把所有鸡蛋砸碎。 *(The boy said that after he refused to hand over 100 rupees, the officials overturned his car and smashed all the eggs.)*
Best of five translation	当他拒绝时，官员将他的车子推倒，破坏了所有的蛋，男孩称。 *(When he refused, officials pushed his car over and broke all the eggs, the boy said.)*

Table E.1: Two additional examples showing the reference translations can be less accurate than the best model prediction.

E.2.4 *More Examples*

Table E.1 shows two additional examples in which the model's translation scores higher than the reference translation. This once again highlights the presence of noise in parallel datasets.

E.3 MORE IMPLEMENTATION DETAILS

E.3.1 *Dataset statistics*

The data statistics are presented in Table E.2. We use different validation sets in different training stages because MAPLE contains a subset of the parallel data in WMT20/21.

E.3.2 *Prompt format*

For each source sentence, we attach a MT instruction asking the LLM to generate the translation. The MT instructions come from a instruction

	Training stage	Data source	Number of samples			
			de→en	en→de	en→zh	zh→en
Training	SFT stage	WMT17	3004	3004	2001	2001
		WMT18	2998	2998	3981	3981
		WMT19	2000	1997	1997	2000
	PL stage	MAPLE	1100	1100	1100	1100
Validation	SFT stage	WMT21	1000	1002	1002	1948
	PL stage	WMT20 & 21*	500	500	500	500
Test	-	WMT22	1984	2037	2037	1875
		FLORES-200	1012	1012	1012	1012
Preference testing	-	MAPLE-dev	217	195	208	180

Table E.2: Datasets used for training, validation and testing. *: a subset WMT20 and WMT21 is used.

pool based on the list of MT instructions released by (Jiao et al., 2023a)[1]. We list all 31 instructions in our instruction pool in Table E.3. During training (in both SFT and PL stages), an instruction is randomly sampled from the instruction pool. During evaluation, the first instruction from Table E.3 is always used. In addition to instructions, instruction-tuned models like Vicuna requires specific prompt formats. Table E.4 presents a depiction of the conversion process from raw data points to the final model input.

E.3.3 *Hyper-parameter search*

Hyper-parameter search is done for $\beta \in [0.0, 0.05, 0.1]$, and best values are selected according to the validation loss.

E.3.4 *Evaluation packages*

We use the `Unbabel/wmt22-comet-da` model[2] to compute the COMET scores and sacreBLEU[3] for computing BLEU scores. The signature of the sacreBLEU package is `nrefs:1, case:mixed, eff:no, tok:13a, smooth:exp, version:2.0.0` for all translation directions but `en→zh`, in which we use `tok:zh`.

E.3.5 *Hardware specifications and runtime*

All experiments are either run on a host with eight NVIDIA A100-40GB GPUs or with eight H100-80GB GPUs. Mixed precision with

1 https://github.com/wxjiao/ParroT
2 https://github.com/Unbabel/COMET
3 https://github.com/mjpost/sacrebleu

Instruction pool
Translate the following text from [SRC] to [TGT]:
Please provide the [TGT] translation for the following text
Convert the subsequent sentences from [SRC] into [TGT]:
Render the listed sentences in [TGT] from their original [SRC] form:
Transform the upcoming sentences from [SRC] language to [TGT] language:
Translate the given text from [SRC] to [TGT]:
Turn the following sentences from their [SRC] version to the [TGT] version:
Adapt the upcoming text from [SRC] to [TGT]:
Transpose the next sentences from the [SRC] format to the [TGT] format.
Reinterpret the ensuing text from [SRC] to [TGT] language.
Modify the forthcoming sentences, converting them from [SRC] to [TGT].
What is the meaning of these sentences when translated to [TGT]?
In the context of [TGT], what do the upcoming text signify? The text is:
How would you express the meaning of the following sentences in [TGT]?
What is the significance of the mentioned sentences in [TGT]?
In [TGT], what do the following text convey?
When translated to [TGT], what message do these sentences carry?
What is the intended meaning of the ensuing sentences in [TGT]?
How should the following sentences be comprehended in [TGT]?
In terms of [TGT], what do the next sentences imply?
Kindly furnish the [TGT] translation of the subsequent sentences.
Could you supply the [TGT] translation for the upcoming sentences?
Please offer the [TGT] rendition for the following statements.
I'd appreciate it if you could present the [TGT] translation for the following text:
Can you deliver the [TGT] translation for the mentioned sentences?
Please share the [TGT] version of the given sentences.
It would be helpful if you could provide the [TGT] translation of the ensuing sentences.
Kindly submit the [TGT] interpretation for the next sentences.
Please make available the [TGT] translation for the listed sentences.
Can you reveal the [TGT] translation of the forthcoming sentences?
Translate from [SRC] to [TGT]:

Table E.3: An instruction pool containing 31 MT prompts. An instruction is randomly sampled from this pool to form a training sample. At inference time, the first instruction is always used. [SRC] and [TGT] will be replaced by the source and target language, respectively.

Model	Instruction template
Vicuna	`USER: [MT Instruction] \nASSISTANT:\n`
Mistral-Instruct	`[INST] [MT Instruction] \n[\INST]`
BLOOMZ	`USER: [MT Instruction] \nASSISTANT:\n`

Example

`USER: Translate the following text from English to German: Hello, world. \nASSISTANT:\n Hallo, Welt.`

Table E.4: (a) Instruction template used for Vicuna, Mistral-Instruct, and BLOOMZ. Raw template is marked in red. BLOOMZ shares the same template as Vicuna at the SFT and PL stage. When performing BLOOMZ on zero-shot tasks, we directly use the first instruction from Table E.3 without any instruction template. (b) An example that converts the raw input (marked in green) to the final input.

bfloat16 is used in both SFT and PL. Deepspeed[4] zero-stage 3 is used when running PL with five preference samples. Each experiment runs no longer than 15 minutes on H100 GPUs.

E.4 SFT RESULTS IN BLEU SCORE

We present model performance after SFT stage measured by BLEU score in Table E.5. While the general trend remains consistent in comparison to the performance evaluated by COMET, there are some exceptions. For example, although VicunaMT still achieves the top average score on FLORES-200, it is outperformed by MistralMT (i.e., Mistral + SFT) on WMT22.

E.5 MODEL COMPARISON IN BLEU SCORE

We present model performance measured by BLEU score in Table E.6. In this case, there is no clear winner. Interestingly, VicunaMT+PL attains lower BLEU scores than VicunaMT on `en`→`de` and `zh`→`en` when evaluated on WMT22. However, both COMET score and our human evaluation in Table 7.4 show the opposite, highlighting that BLEU scores may less correlated to human judgement, as also noticed in (Freitag et al., 2022).

4 https://github.com/microsoft/DeepSpeed

	de→en	en→de	en→zh	zh→en	Avg.
			WMT22		
BLOOM	1.51	0.53	1.74	5.43	2.30
+*SFT*	23.73	16.15	35.15	21.64	24.17
BLOOMZ	21.59	6.79	28.72	18.54	18.91
+*SFT*	23.89	16.79	**35.41**	21.01	24.28
Mistral	4.32	2.65	4.93	7.01	4.73
+*SFT*	**29.39**	24.60	31.51	**22.09**	**26.90**
Mistral-Ins.	28.04	21.27	21.85	17.77	22.23
+*SFT*	28.26	24.61	31.90	20.60	26.35
LLaMA-1	6.30	4.00	0.88	3.01	3.55
+*SFT*	28.28	19.09	25.31	20.27	23.24
Vicuna	26.16	22.11	26.26	13.91	22.11
+*SFT*	29.26	**25.70**	29.98	20.61	26.39
			FLORES-200		
BLOOM	3.88	1.48	7.00	3.75	4.03
+*SFT*	31.85	16.26	**34.66**	**23.78**	26.64
Mistral	3.58	1.37	0.16	1.06	1.54
+*SFT*	40.48	29.18	29.43	24.67	30.94
Mistral-Ins.	36.81	25.64	19.81	19.25	25.38
+*SFT*	39.16	27.79	29.77	23.10	29.96
LLaMA-1	4.08	2.80	1.73	1.60	2.55
+*SFT*	40.70	29.95	20.21	20.66	27.88
Vicuna	35.07	26.86	26.09	17.53	26.39
+*SFT*	**41.90**	**30.63**	28.52	23.34	**31.10**

Table E.5: Model performance (in BLEU score) before and after performing SFT on parallel data. Rows in blue indicate instruction-tuned LLMs. Best results are in **bold**. Instruction-tuned LLMs perform well even without SFT. Raw LLMs benefits the most from SFT. We exclude BLOOMZ on FLORES-200 as it is a part of BLOOMZ's training data.

E.6 DATA REUSE IN BLEU SCORE AND RESULTS ON FLORES-200

We reuse MAPLE to enhance BLOOMZMT and MistralInstructMT (i.e., BLOOMZ and MistralInstruct after the SFT stage) and report model performance on WMT22 in BLEU score in Table E.7. In addition, we evaluate MistralInstructMT on FLORES and present the results in Table E.8.

System	WMT22					FLORES-200				
	de→en	en→de	en→zh	zh→en	Avg.	de→en	en→de	en→zh	zh→en	Avg.
Commercial & LLaMA-2-7B based MT systems										
ChatGPT$_{(3.5\text{-turbo-}0613)}$	33.13	33.56	44.59	25.63	31.62	43.06	40.07	45.69	25.57	36.55
GPT-4$_{(gpt\text{-}4\text{-}0613)}$	33.72	34.84	42.75	26.33	34.41	43.79	41.81	46.10	27.39	39.77
ALMA-7B$_{(LLaMA\text{-}2)}$	29.49	30.31	36.48	23.52	29.95	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$	-$^{\otimes}$
BLOOMZ-mt-7B based LLMs										
ParroT$_{(BLOOMZ\text{-}mt)}$	24.90	20.50	34.50	22.70	25.65	-*	-*	-*	-*	-*
TIM$_{(BLOOMZ\text{-}mt)}$	24.31	20.63	37.20	23.42	26.39	-*	-*	-*	-*	-*
SWIE$_{(BLOOMZ\text{-}mt)}$	25.95	21.83	36.88	23.33	27.00	-*	-*	-*	-*	-*
LLaMA-1-7B based LLMs										
ParroT$_{(LLaMA\text{-}1)}$	27.30	26.10	30.30	20.20	25.98	39.40	30.70	29.10	21.30	32.38
TIM$_{(LLaMA\text{-}1)}$	27.91	25.02	30.07	19.33	25.58	39.15	29.31	28.43	22.30	29.80
SWIE$_{(LLaMA\text{-}1)}$	30.48	**27.10**	31.08	**21.19**	**27.47**	40.20	**31.41**	29.07	21.59	30.57
VicunaMT$_{(LLaMA\text{-}1)}$	29.26	25.70	29.98	20.61	26.39	**41.90**	30.63	28.52	**23.34**	**31.10**
+ REF	**31.12**	24.72	30.07	20.38	26.58	39.03	29.36	28.87	22.84	30.03
+ BEST	29.44	24.93	30.91	20.39	26.16	41.29	29.34	30.07	23.48	31.05
+ PL	30.63	24.63	**31.52**	20.44	26.81	40.07	29.33	**30.50**	21.99	30.47

Table E.6: Model performance in BLEU scores. Best results with LLaMA-1 based models are in **bold**. $^{\otimes}$: LLaMA-2 based models were not evaluated due to license constraints. WMT22 results are extracted from the original paper. *: BLOOMZ-family models use FLORES-200 for training.

	WMT22				
	de→en	en→de	en→zh	zh→en	Avg.
BLOOMZ†	23.89	16.79	35.41	21.01	24.28
+REF	24.51	15.26	33.43	21.80	23.75
+BEST	23.80	16.33	34.99	21.49	24.15
+PL	**24.84**	**16.81**	**36.48**	**23.15**	**25.32**
Mistral-Ins.†	28.26	24.61	31.90	20.60	26.35
+REF	**30.94**	**25.62**	31.66	21.52	27.44
+BEST	29.76	24.30	31.12	20.83	26.50
+PL	29.32	24.78	**33.00**	**21.76**	**27.47**

Table E.7: Model performance on WMT22 in BLEU scores. Best results are in **bold**. †: SFT stage has already been applied to these models.

	FLORES-200				
	de→en	en→de	en→zh	zh→en	Avg.
			COMET		
Mistral-Ins.†	88.21	83.73	82.41	84.77	84.78
+REF	88.10	85.04	83.59	84.74	85.37
+BEST	88.41	84.55	83.46	84.94	85.34
+PL	**88.56**	**84.98**	**83.86**	**85.34**	**85.67**
			BLEU		
Mistral-Ins.†	39.16	27.79	29.77	23.10	29.96
+REF	38.10	**28.39**	**31.24**	23.09	30.21
+BEST	39.35	28.33	30.46	22.98	30.28
+PL	**39.80**	27.97	31.00	**23.44**	**30.55**

Table E.8: Model performance on FLORES-200 in COMET and BLEU scores. Best results are in **bold**. †: SFT stage has already been applied to these models.

BIBLIOGRAPHY

Agrawal, Sweta, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad (July 2023). "In-context Examples Selection for Machine Translation." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 8857–8873. DOI: 10.18653/v1/2023.findings-acl.564. URL: https://aclanthology.org/2023.findings-acl.564.

Akhbardeh, Farhad et al. (Nov. 2021). "Findings of the 2021 Conference on Machine Translation (WMT21)." In: *Proceedings of the Sixth Conference on Machine Translation*. Ed. by Loic Barrault et al. Online: Association for Computational Linguistics, pp. 1–88. URL: https://aclanthology.org/2021.wmt-1.1.

Alex, Neel et al. (2021). "RAFT: A Real-World Few-Shot Text Classification Benchmark." In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*. Ed. by Joaquin Vanschoren and Sai-Kit Yeung.

Algan, Görkem and Ilkay Ulusoy (2021). "Image classification with deep learning in the presence of noisy labels: A survey." In: *Knowl. Based Syst.* 215, p. 106771. DOI: 10.1016/j.knosys.2021.106771. URL: https://doi.org/10.1016/j.knosys.2021.106771.

Alves, Duarte M., Nuno M. Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins (2023a). "Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning." In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. URL: https://aclanthology.org/2023.findings-emnlp.744/.

Alves, Duarte M., Nuno Miguel Guerreiro, João Alves, José Pombal, Ricardo Rei, José G. C. de Souza, Pierre Colombo, and André F. T. Martins (2023b). "Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning." In: *CoRR* abs/2310.13448. DOI: 10.48550/ARXIV.2310.13448. arXiv: 2310.13448. URL: https://doi.org/10.48550/arXiv.2310.13448.

Alves, Duarte M., José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. (2024). "Tower: An Open Multilingual Large Language Model for Translation-Related Tasks." In: *arXiv preprint*. URL: https://arxiv.org/abs/2402.17733.

Angluin, Dana and Philip Laird (1988). "Learning from noisy examples." In: *Machine Learning* 2.4, pp. 343–370.

Arpit, Devansh et al. (2017). "A Closer Look at Memorization in Deep Networks." In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 233–242. URL: http://proceedings.mlr.press/v70/arpit17a.html.

Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama (July 2020). "On the Cross-lingual Transferability of Monolingual Representations." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, pp. 4623–4637. DOI: 10.18653/v1/2020.acl-main.421. URL: https://aclanthology.org/2020.acl-main.421.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi (2023). *Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection*. arXiv: 2310.11511 [cs.CL]. URL: https://arxiv.org/abs/2310.11511.

Awasthi, Abhijeet, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi (2020). "Learning from Rules Generalizing Labeled Exemplars." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=SkeuexBtDr.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv: 1607.06450 [stat.ML]. URL: https://arxiv.org/abs/1607.06450.

Barrault, Loïc et al. (Aug. 2019). "Findings of the 2019 Conference on Machine Translation (WMT19)." In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Ed. by Ondřej Bojar et al. Florence, Italy: Association for Computational Linguistics, pp. 1–61. DOI: 10.18653/v1/W19-5301. URL: https://aclanthology.org/W19-5301.

Barrault, Loïc et al. (Nov. 2020). "Findings of the 2020 Conference on Machine Translation (WMT20)." In: *Proceedings of the Fifth Conference on Machine Translation*. Ed. by Loïc Barrault et al. Online: Association for Computational Linguistics, pp. 1–55. URL: https://aclanthology.org/2020.wmt-1.1.

Bawden, Rachel and François Yvon (2023). "Investigating the Translation Performance of a Large Multilingual Language Model: the Case of BLOOM." In: *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*. Ed. by Mary Nurminen et al. European Association for Machine Translation, pp. 157–170. URL: https://aclanthology.org/2023.eamt-1.16.

Bekker, Alan Joseph and Jacob Goldberger (2016). "Training deep neural-networks based on unreliable labels." In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*

2016, Shanghai, China, March 20-25, 2016. IEEE, pp. 2682–2686. DOI: 10.1109/ICASSP.2016.7472164. URL: https://doi.org/10.1109/ICASSP.2016.7472164.

Berthelot, David, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel (2019). "MixMatch: A Holistic Approach to Semi-Supervised Learning." 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*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 5050–5060. URL: https://proceedings.neurips.cc/paper/2019/hash/1cd138d0499a68f4bb72bee04bbec2d7-Abstract.html.

Bi, Wei, Liwei Wang, James T. Kwok, and Zhuowen Tu (2014). "Learning to Predict from Crowdsourced Data." In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*. Ed. by Nevin L. Zhang and Jin Tian. AUAI Press, pp. 82–91. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1\&smnu=2\&article_id=2443\&proceeding_id=30.

Bojar, Ondřej, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz (Oct. 2018). "Findings of the 2018 Conference on Machine Translation (WMT18)." In: *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Ed. by Ondřej Bojar et al. Belgium, Brussels: Association for Computational Linguistics, pp. 272–303. DOI: 10.18653/v1/W18-6401. URL: https://aclanthology.org/W18-6401.

Bojar, Ondřej et al. (Sept. 2017). "Findings of the 2017 Conference on Machine Translation (WMT17)." In: *Proceedings of the Second Conference on Machine Translation*. Ed. by Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer. Copenhagen, Denmark: Association for Computational Linguistics, pp. 169–214. DOI: 10.18653/v1/W17-4717. URL: https://aclanthology.org/W17-4717.

Bradley, Ralph Allan (1953). "Some Statistical Methods in Taste Testing and Quality Evaluation." In: *Biometrics* 9.1, pp. 22–38. ISSN: 0006-341X. DOI: 10.2307/3001630. URL: https://www.jstor.org/stable/3001630.

Bradley, Ralph Allan and Milton E Terry (1952). "Rank analysis of incomplete block designs: I. The method of paired comparisons." In: *Biometrika* 39.3/4, pp. 324–345.

Bragg, Jonathan, Arman Cohan, Kyle Lo, and Iz Beltagy (2021). "FLEX: Unifying Evaluation for Few-Shot NLP." In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021,*

virtual. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, pp. 15787–15800. URL: https://proceedings.neurips.cc/paper/2021/hash/8493eeaccb772c0878f99d60a0bd2bb3-Abstract.html.

Briakou, Eleftheria, Colin Cherry, and George Foster (July 2023). "Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM's Translation Capability." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 9432–9452. DOI: 10.18653/v1/2023.acl-long.524. URL: https://aclanthology.org/2023.acl-long.524.

Brown, Tom et al. (2020). "Language Models are Few-Shot Learners." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 1877–1901.

Burns, Collin et al. (2023). *Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision*. arXiv: 2312.09390 [cs.CL]. URL: https://arxiv.org/abs/2312.09390.

Chen, Lichang, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. (2024a). "Alpagasus: Training a Better Alpaca Model with Fewer Data." In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/pdf?id=FdVXgSJhvz.

Chen, Pengfei, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng (2020). "Robustness of Accuracy Metric and its Inspirations in Learning with Noisy Labels." In: *CoRR* abs/2012.04193. arXiv: 2012.04193. URL: https://arxiv.org/abs/2012.04193.

– (2021). "Beyond Class-Conditional Assumption: A Primary Attempt to Combat Instance-Dependent Label Noise." 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*. AAAI Press, pp. 11442–11450. DOI: 10.1609/AAAI.V35I13.17363. URL: https://doi.org/10.1609/aaai.v35i13.17363.

Chen, Pinzhen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield (2024b). "Iterative Translation Refinement with Large Language Models." In: *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*. URL: https://aclanthology.org/2024.eamt-1.17.

Chen, Pinzhen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield (2024c). "Monolingual or Multilingual Instruction Tuning: Which Makes a Better Alpaca." In: *Findings of the Association for Computational Linguistics: EACL 2024*. URL: https://aclanthology.org/2024.findings-eacl.90.

Chen, Xinlei and Abhinav Gupta (2015). "Webly supervised learning of convolutional networks." In: *Proceedings of the IEEE international conference on computer vision*, pp. 1431–1439.

Chen, Xinlei, Abhinav Shrivastava, and Abhinav Gupta (2013). "NEIL: Extracting Visual Knowledge from Web Data." In: *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. IEEE Computer Society, pp. 1409–1416. DOI: 10.1109/ICCV.2013.178. URL: https://doi.org/10.1109/ICCV.2013.178.

Chen, Yijie, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou (2023). "Improving Translation Faithfulness of Large Language Models via Augmenting Instructions." In: *arXiv preprint arXiv:2308.12674*.

Cheng, Weiwei, Krzysztof Dembczynski, and Eyke Hüllermeier (2010). "Label Ranking Methods based on the Plackett-Luce Model." In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. Ed. by Johannes Fürnkranz and Thorsten Joachims. Omnipress, pp. 215–222. URL: https://icml.cc/Conferences/2010/papers/353.pdf.

Cheng, Weiwei and Eyke Hüllermeier (2008). "Learning similarity functions from qualitative feedback." In: *LNAI 5239 Advances in Case-Based Reasoning: The 9th European Conference on Case-Based Reasoning (ECCBR-08)*. Ed. by Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft. Trier, Germany: Springer, pp. 129–134.

Chiang, Wei-Lin, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. (2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. lmsys.org. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

Chirkova, Nadezhda and Vassilina Nikoulina (2024a). "Key ingredients for effective zero-shot cross-lingual knowledge transfer in generative tasks." In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. DOI: 10.18653/v1/2024.naacl-long.401.

– (2024b). "Zero-shot cross-lingual transfer in instruction tuning of large language models." In: *Proceedings of the 17th International Natural Language Generation Conference*. URL: https://aclanthology.org/2024.inlg-main.53.

Choenni, Rochelle, Dan Garrette, and Ekaterina Shutova (2023). "How do languages influence each other? Studying cross-lingual data sharing during LM fine-tuning." In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. URL: https://aclanthology.org/2023.emnlp-main.818/.

Chowdhery, Aakanksha et al. (2023). "PaLM: Scaling Language Modeling with Pathways." In: *Journal of Machine Learning Research* 24.240, pp. 1–113. URL: http://jmlr.org/papers/v24/22-1144.html.

Chung, Hyung Won, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. (2024). "Scaling Instruction-Finetuned Language Models." In: *Journal of Machine Learning Research*. URL: http://jmlr.org/papers/v25/23-0870.html.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning (2020). "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=r1xMH1BtvB.

Costa-jussà, Marta R. et al. (2022). "No Language Left Behind: Scaling Human-Centered Machine Translation." In: *CoRR* abs/2207.04672. DOI: 10.48550/ARXIV.2207.04672. arXiv: 2207.04672. URL: https://doi.org/10.48550/arXiv.2207.04672.

Davis, Allan Peter et al. (2013). "A CTD-Pfizer collaboration: manual curation of 88 000 scientific articles text mined for drug-disease and drug-phenotype interactions." In: *Database J. Biol. Databases Curation* 2013. DOI: 10.1093/database/bat080. URL: https://doi.org/10.1093/database/bat080.

Dehghani, Mostafa, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf (2018). "Fidelity-Weighted Learning." In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=B1X0mzZCW.

Dehghani, Mostafa, Aliaksei Severyn, Sascha Rothe, and Jaap Kamps (2017). "Avoiding Your Teacher's Mistakes: Training Neural Networks with Controlled Weak Supervision." In: *CoRR* abs/1711.00313. arXiv: 1711.00313. URL: http://arxiv.org/abs/1711.00313.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423.

Dong, Qingxiu, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei (2024). *Self-Boosting Large Language Models with Synthetic Preference Data*. arXiv: 2410.06961 [cs.CL]. URL: https://arxiv.org/abs/2410.06961.

Dong, Yi, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev (2023). "SteerLM: Attribute Conditioned SFT as an (User-Steerable) Alternative to RLHF." In: *CoRR* abs/2310.05344.

DOI: 10.48550/ARXIV.2310.05344. arXiv: 2310.05344. URL: https://doi.org/10.48550/arXiv.2310.05344.

Eberhard, David M., Gary F. Simons, and Charles D. Fennig (eds.) (2019). *Ethnologue: Languages of the World. Twenty-second edition.* SIL International. URL: http://www.ethnologue.com.

Fan, Jianping, Yi Shen, Ning Zhou, and Yuli Gao (June 2010). "Harvesting large-scale weakly-tagged image databases from the web." In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 802–809. DOI: 10.1109/cvpr.2010.5540135.

Freitag, Markus, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins (Dec. 2022). "Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust." In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 46–68. URL: https://aclanthology.org/2022.wmt-1.2.

Frénay, Benoît and Ata Kabán (2014). "A comprehensive introduction to label noise." In: *22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014*. URL: https://www.esann.org/sites/default/files/proceedings/legacy/es2014-10.pdf.

Fu, Daniel, Mayee Chen, Frederic Sala, Sarah Hooper, Kayvon Fatahalian, and Christopher Ré (2020). "Fast and three-rious: Speeding up weak supervision with triplet methods." In: *International Conference on Machine Learning*. PMLR, pp. 3280–3291.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by Maria-Florina Balcan and Kilian Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html.

Gao, Tianyu, Adam Fisch, and Danqi Chen (2021). "Making Pre-trained Language Models Better Few-shot Learners." In: *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*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Association for Computational Linguistics, pp. 3816–3830. DOI: 10.18653/v1/2021.acl-long.295. URL: https://doi.org/10.18653/v1/2021.acl-long.295.

Ghazvininejad, Marjan, Hila Gonen, and Luke Zettlemoyer (2023a). "Dictionary-based Phrase-level Prompting of Large Language Models for Machine Translation." In: *arXiv preprint*. URL: https://arxiv.org/abs/2302.07856.

Ghazvininejad, Marjan, Hila Gonen, and Luke Zettlemoyer (2023b). "Dictionary-based Phrase-level Prompting of Large Language Models for Machine Translation." In: *CoRR* abs/2302.07856. DOI: 10.48550/ARXIV.2302.07856. arXiv: 2302.07856. URL: https://doi.org/10.48550/arXiv.2302.07856.

Ghosh, Aritra, Himanshu Kumar, and P. S. Sastry (2017). "Robust Loss Functions under Label Noise for Deep Neural Networks." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, pp. 1919–1925. DOI: 10.1609/AAAI.V31I1.10894. URL: https://doi.org/10.1609/aaai.v31i1.10894.

Ghosh, Aritra and Andrew S. Lan (2021). "Do We Really Need Gold Samples for Sample Weighting under Label Noise?" In: *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, pp. 3921–3930. DOI: 10.1109/WACV48630.2021.00397. URL: https://doi.org/10.1109/WACV48630.2021.00397.

Ghosh, Aritra, Naresh Manwani, and P. S. Sastry (2015). "Making risk minimization tolerant to label noise." In: *Neurocomputing* 160, pp. 93–107. DOI: 10.1016/J.NEUCOM.2014.09.081. URL: https://doi.org/10.1016/j.neucom.2014.09.081.

Ghosh, Sreyan, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha (2024). "A Closer Look at the Limitations of Instruction Tuning." In: *Proceedings of the 41st International Conference on Machine Learning*. URL: https://proceedings.mlr.press/v235/ghosh24a.html.

Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli (2023). "ChatGPT outperforms crowd workers for text-annotation tasks." In: *Proceedings of the National Academy of Sciences* 120.30, e2305016120. DOI: 10.1073/pnas.2305016120. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2305016120. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2305016120.

Goldberger, Jacob and Ehud Ben-Reuven (2017). "Training deep neural-networks using a noise adaptation layer." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=H12GRgcxg.

Grattafiori, Aaron et al. (2024). *The Llama 3 Herd of Models*. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

Gu, Keren, Xander Masotto, Vandana Bachani, Balaji Lakshminarayanan, Jack Nikodem, and Dong Yin (2021). "An instance-dependent simulation framework for learning with label noise." In: *arXiv preprint arXiv:2107.11413*.

Guan, Melody Y., Varun Gulshan, Andrew M. Dai, and Geoffrey E. Hinton (2018). "Who Said What: Modeling Individual Labelers

Improves Classification." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 3109–3118. DOI: 10.1609/AAAI.V32I1.11756. URL: https://doi.org/10.1609/aaai.v32i1.11756.

Guo, Daya, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. (2025). "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning." In: *arXiv preprint arXiv:2501.12948*.

Hamilton, Ian, Nick Tawn, and David Firth (Dec. 2023). "The many routes to the ubiquitous Bradley-Terry model." In: arXiv:2312.13619. DOI: 10.48550/arXiv.2312.13619. URL: http://arxiv.org/abs/2312.13619.

Han, Bo, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor W. Tsang, and Masashi Sugiyama (2020). "SIGUA: Forgetting May Make Learning with Noisy Labels More Robust." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4006–4016. URL: http://proceedings.mlr.press/v119/han20c.html.

Han, Bo, Ivor W. Tsang, and Ling Chen (2016). "On the Convergence of a Family of Robust Losses for Stochastic Gradient Descent." In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I*. Ed. by Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken. Vol. 9851. Lecture Notes in Computer Science. Springer, pp. 665–680. DOI: 10.1007/978-3-319-46128-1_42. URL: https://doi.org/10.1007/978-3-319-46128-1_42.

Han, Bo, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor W. Tsang, Ya Zhang, and Masashi Sugiyama (2018a). "Masking: A New Perspective of Noisy Supervision." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 5841–5851. URL: https://proceedings.neurips.cc/paper/2018/hash/aee92f16efd522b9326c25cc3237ac15-Abstract.html.

Han, Bo, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama (2018b). "Co-teaching: Robust training of deep neural networks with extremely noisy labels." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*

2018, December 3-8, 2018, Montréal, Canada. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 8536–8546.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen (2021). "Deberta: decoding-Enhanced Bert with Disentangled Attention." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=XPZIaotutsD.

He, Zhiwei, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang (2023). "Exploring Human-Like Translation Strategy with Large Language Models." In: *CoRR* abs/2305.04118. DOI: 10.48550/ARXIV.2305.04118. arXiv: 2305.04118. URL: https://doi.org/10.48550/arXiv.2305.04118.

– (2024). "Exploring Human-Like Translation Strategy with Large Language Models." In: *Transactions of the Association for Computational Linguistics*. DOI: 10.1162/tacl_a_00642.

Hedderich, Michael A., David Ifeoluwa Adelani, Dawei Zhu, Jesujoba O. Alabi, Udia Markus, and Dietrich Klakow (2020). "Transfer Learning and Distant Supervision for Multilingual Transformer Models: A Study on African Languages." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, pp. 2580–2591. DOI: 10.18653/v1/2020.emnlp-main.204. URL: https://doi.org/10.18653/v1/2020.emnlp-main.204.

Hedderich, Michael A., Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow (June 2021). "A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 2545–2568. DOI: 10.18653/v1/2021.naacl-main.201. URL: https://aclanthology.org/2021.naacl-main.201.

Hedderich, Michael A., Dawei Zhu, and Dietrich Klakow (2021). "Analysing the Noise Model Error for Realistic Noisy Label Data." In: *To appear at AAAI 2021*.

Hejna, Joey, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh (2023). "Contrastive Preference Learning: Learning from Human Feedback without RL." In: *CoRR* abs/2310.13639. DOI: 10.48550/ARXIV.2310.13639. arXiv: 2310.13639. URL: https://doi.org/10.48550/arXiv.2310.13639.

Hendrickx, Iris, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz (July 2010). "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of

Nominals." In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pp. 33–38. URL: https://aclanthology.org/S10-1006.

Hendrycks, Dan, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel (2018). "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 10477–10486. URL: https://proceedings.neurips.cc/paper/2018/hash/ad554d8c3b06d6b97ee76a2448bd7913-Abstract.html.

Hendy, Amr, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla (2023). "How good are GPT models at machine translation? a comprehensive evaluation." In: *arXiv preprint arXiv:2302.09210*.

Herold, Christian, Jan Rosendahl, Joris Vanvinckenroye, and Hermann Ney (2022). "Detecting Various Types of Noise for Neural Machine Translation." In: *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 2542–2551. DOI: 10.18653/V1/2022.FINDINGS-ACL.200. URL: https://doi.org/10.18653/v1/2022.findings-acl.200.

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi (2020). "The Curious Case of Neural Text Degeneration." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rygGQyrFvH.

Honovich, Or, Thomas Scialom, Omer Levy, and Timo Schick (2022). *Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor*. arXiv: 2212.09689 [cs.CL]. URL: https://arxiv.org/abs/2212.09689.

Houle, Michael E. (2017). "Local intrinsic dimensionality I: an extreme-value-theoretic foundation for similarity applications." In: *Proceedings of SISAP*.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). "Parameter-Efficient Transfer Learning for NLP." In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2790–2799. URL: http://proceedings.mlr.press/v97/houlsby19a.html.

Howard, Jeremy and Sebastian Ruder (2018). "Universal Language Model Fine-tuning for Text Classification." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://aclanthology.org/P18-1031/.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen (2022). "LoRA: Low-Rank Adaptation of Large Language Models." In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, Jian, Li Tao, June Yang, and Chandler Zhou (2023). "Aligning Language Models with Offline Reinforcement Learning from Human Feedback." In: *CoRR* abs/2308.12050. DOI: 10.48550/ARXIV.2308.12050. arXiv: 2308.12050. URL: https://doi.org/10.48550/arXiv.2308.12050.

Hu, Wei, Zhiyuan Li, and Dingli Yu (2020). "Simple and Effective Regularization Methods for Training on Noisily Labeled Data with Generalization Guarantee." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=Hke3gyHYwH.

Huang, Lang, Chao Zhang, and Hongyang Zhang (2020). "Self-Adaptive Training: beyond Empirical Risk Minimization." In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin.

Huang, Shengyi, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang (2022). "The 37 Implementation Details of Proximal Policy Optimization." In: *ICLR Blog Track*. https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/. URL: https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/.

Iscen, Ahmet, Jack Valmadre, Anurag Arnab, and Cordelia Schmid (2022). "Learning with Neighbor Consistency for Noisy Labels." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 4662–4671. DOI: 10.1109/CVPR52688.2022.00463. URL: https://doi.org/10.1109/CVPR52688.2022.00463.

Islam, Riashat, Peter Henderson, Maziar Gomrokchi, and Doina Precup (2017). "Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control." In: *CoRR* abs/1708.04133. arXiv: 1708.04133. URL: http://arxiv.org/abs/1708.04133.

Iyer, Vivek, Pinzhen Chen, and Alexandra Birch (2023). "Towards Effective Disambiguation for Machine Translation with Large Language Models." In: *Proceedings of the Eighth Conference on Machine Translation*. URL: https://aclanthology.org/2023.wmt-1.44/.

Ji, Jiaming et al. (2024). *AI Alignment: A Comprehensive Survey*. arXiv: 2310.19852 [cs.AI]. URL: https://arxiv.org/abs/2310.19852.

Jiang, Albert Q, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. (2023a). "Mistral 7B." In: *arXiv preprint*. URL: https://arxiv.org/abs/2310.06825.

Jiang, Albert Q. et al. (2023b). "Mistral 7B." In: *CoRR* abs/2310.06825. DOI: 10.48550/ARXIV.2310.06825. arXiv: 2310.06825. URL: https://doi.org/10.48550/arXiv.2310.06825.

Jiang, Haoming, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao (2021). "Named Entity Recognition with Small Strongly Labeled and Large Weakly Labeled Data." In: *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*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Association for Computational Linguistics, pp. 1775–1789. DOI: 10.18653/v1/2021.acl-long.140. URL: https://doi.org/10.18653/v1/2021.acl-long.140.

Jiang, Lu, Di Huang, Mason Liu, and Weilong Yang (2020). "Beyond Synthetic Noise: Deep Learning on Controlled Noisy Labels." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4804–4815. URL: http://proceedings.mlr.press/v119/jiang20c.html.

Jiang, Lu, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei (2018). "MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2309–2318. URL: http://proceedings.mlr.press/v80/jiang18c.html.

Jiao, Wenxiang, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu (2023a). "ParroT: Translating during Chat using Large Language Models tuned with Human Translation and Feedback." In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. DOI: 10.18653/v1/2023.findings-emnlp.1001. URL: https://aclanthology.org/2023.findings-emnlp.1001.

Jiao, Wenxiang, Wenxuan Wang, JT Huang, Xing Wang, and ZP Tu (2023b). "Is ChatGPT a good translator? Yes with GPT-4 as the engine." In: *arXiv preprint arXiv:2301.08745*.

Jiao, Xiaoqi, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu (2020). "TinyBERT: Distilling BERT for Natural Language Understanding." In: *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Vol. EMNLP 2020. Findings of ACL. Association for Computational Linguistics, pp. 4163–4174. DOI: 10.18653/V1/2020.FINDINGS-EMNLP.372. URL: https://doi.org/10.18653/v1/2020.findings-emnlp.372.

Jindal, Ishan, Matthew S. Nokleby, and Xue-wen Chen (2017). "Learning Deep Networks from Noisy Labels with Dropout Regularization." In: *CoRR* abs/1705.03419. arXiv: 1705.03419. URL: http://arxiv.org/abs/1705.03419.

Jindal, Ishan, Matthew S. Nokleby, and Daniel Pressel (2019). "A Nonlinear, Noise-aware, Quasi-clustering Approach to Learning Deep CNNs from Noisy Labels." In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 64–72.

Jindal, Ishan, Daniel Pressel, Brian Lester, and Matthew S. Nokleby (2019). "An Effective Label Noise Model for DNN Text Classification." In: *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)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 3246–3256. DOI: 10.18653/v1/n19-1328. URL: https://doi.org/10.18653/v1/n19-1328.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). "Scaling Laws for Neural Language Models." In: *CoRR* abs/2001.08361. arXiv: 2001.08361. URL: https://arxiv.org/abs/2001.08361.

Karamanolakis, Giannis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah (2021). "Self-Training with Weak Supervision." In: *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*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Association for Computational Linguistics, pp. 845–863. DOI: 10.18653/v1/2021.naacl-main.66. URL: https://doi.org/10.18653/v1/2021.naacl-main.66.

Karim, Nazmul, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah (2022). "UNICON: Combating Label Noise Through Uniform Selection and Contrastive Learning." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 9666–9676. DOI: 10.1109/CVPR52688.2022.00945. URL: https://doi.org/10.1109/CVPR52688.2022.00945.

Kew, Tannon, Florian Schottmann, and Rico Sennrich (2023). "Turning English-centric LLMs Into Polyglots: How Much Multilinguality Is Needed?" In: *arXiv preprint*. URL: https://arxiv.org/abs/2312.12683.

Khayrallah, Huda and Philipp Koehn (2018). "On the Impact of Various Types of Noise on Neural Machine Translation." In: *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*. Ed. by Alexandra Birch, Andrew M. Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. Association for Computational Linguistics, pp. 74–83. DOI: 10.18653/V1/W18-2709. URL: https://doi.org/10.18653/v1/w18-2709.

Kocmi, Tom et al. (Dec. 2022). "Findings of the 2022 Conference on Machine Translation (WMT22)." In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 1–45. URL: https://aclanthology.org/2022.wmt-1.1.

Krallinger, Martin, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, José Antonio López, Umesh Nandal, et al. (2017). "Overview of the BioCreative VI chemical-protein interaction Track." In: *Proceedings of the sixth BioCreative challenge evaluation workshop*. Vol. 1, pp. 141–146. URL: https://biocreative.bioinformatics.udel.edu/tasks/biocreative-vi/track-5/.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=H1eA7AEtvS.

Lange, Lukas, Michael A. Hedderich, and Dietrich Klakow (Nov. 2019). "Feature-Dependent Confusion Matrices for Low-Resource NER Labeling with Noisy Labels." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3554–3559. DOI: 10.18653/v1/D19-1362. URL: https://www.aclweb.org/anthology/D19-1362.

Lee, Dong-Hyun et al. (2013). "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." In: *Workshop on challenges in representation learning, ICML*. Vol. 3. 2, p. 896.

Lee, Kuang-Huei, Xiaodong He, Lei Zhang, and Linjun Yang (2018). "CleanNet: Transfer Learning for Scalable Image Classifier Training With Label Noise." In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 5447–5456. DOI: 10.1109/CVPR.2018.00571. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Lee_CleanNet_Transfer_Learning_CVPR_2018_paper.html.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (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*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics, pp. 7871–7880. DOI: 10.18653/V1/2020.ACL-MAIN.703. URL: https://doi.org/10.18653/v1/2020.acl-main.703.

Li, Haonan, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin (2023). "Bactrian-X: Multilingual Replicable Instruction-Following Models with Low-Rank Adaptation." In: *arXiv preprint*. URL: https://arxiv.org/abs/2305.15011.

Li, Jiahuan, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen (2024). "Eliciting the Translation Ability of Large Language Models via Multilingual Finetuning with Translation Instructions." In: *Transactions of the Association for Computational Linguistics*. DOI: 10.1162/tacl_a_00655.

Li, Junnan, Richard Socher, and Steven C. H. Hoi (2020). "DivideMix: Learning with Noisy Labels as Semi-supervised Learning." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=HJgExaVtwr.

Li, Junnan, Caiming Xiong, and Steven C. H. Hoi (2021). "CoMatch: Semi-supervised Learning with Contrastive Graph Regularization." In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, pp. 9455–9464. DOI: 10.1109/ICCV48922.2021.00934. URL: https://doi.org/10.1109/ICCV48922.2021.00934.

Li, Shikun, Xiaobo Xia, Shiming Ge, and Tongliang Liu (2022). "Selective-Supervised Contrastive Learning with Noisy Labels." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, pp. 316–325. DOI:

10.1109/CVPR52688.2022.00041. URL: https://doi.org/10.1109/CVPR52688.2022.00041.

Li, Xin and Dan Roth (2002). "Learning Question Classifiers." In: *COLING 2002: The 19th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C02-1150.

Li, Yuncheng, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li (2017). "Learning from Noisy Labels with Distillation." In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, pp. 1928–1936. DOI: 10.1109/ICCV.2017.211. URL: https://doi.org/10.1109/ICCV.2017.211.

Liang, Chen, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang (2020). "BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision." In: *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. Ed. by Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash. ACM, pp. 1054–1064. DOI: 10.1145/3394486.3403149. URL: https://doi.org/10.1145/3394486.3403149.

Lienen, Julian and Eyke Hüllermeier (2021). "From Label Smoothing to Label Relaxation." 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*. AAAI Press, pp. 8583–8591. DOI: 10.1609/AAAI.V35I10.17041. URL: https://doi.org/10.1609/aaai.v35i10.17041.

Lin, Bill Yuchen, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi (2024). "Urial: Aligning Untuned LLMs with Just the 'Write' Amount of In-Context Learning." In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=wxJ0eXwwda.

Lison, Pierre, Jeremy Barnes, and Aliaksandr Hubin (Aug. 2021). "skweak: Weak Supervision Made Easy for NLP." In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 337–346. DOI: 10.18653/v1/2021.acl-demo.40. URL: https://aclanthology.org/2021.acl-demo.40.

Lison, Pierre, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb (2020). "Named Entity Recognition without Labelled Data: A Weak Supervision Approach." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault. Association for Computational Linguistics,

pp. 1518–1533. DOI: 10.18653/v1/2020.acl-main.139. URL: https://doi.org/10.18653/v1/2020.acl-main.139.

Liu, Aixin, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. (2024a). "Deepseek-v3 technical report." In: *arXiv preprint arXiv:2412.19437*.

Liu, Haokun, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel (2022). "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning." In: *NeurIPS*. URL: http://papers.nips.cc/paper_files/paper/2022/hash/0cde695b83bd186c1fd456302888454c-Abstract-Conference.html.

Liu, Tongliang and Dacheng Tao (2016). "Classification with Noisy Labels by Importance Reweighting." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38.3, pp. 447–461. DOI: 10.1109/TPAMI.2015.2456899. URL: https://doi.org/10.1109/TPAMI.2015.2456899.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *CoRR* abs/1907.11692. arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

Liu, Zuxin et al. (2024b). *APIGen: Automated Pipeline for Generating Verifiable and Diverse Function-Calling Datasets*. arXiv: 2406.18518 [cs.CL]. URL: https://arxiv.org/abs/2406.18518.

Logan IV, Robert, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel (May 2022). "Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models." In: *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, pp. 2824–2835. DOI: 10.18653/v1/2022.findings-acl.222. URL: https://aclanthology.org/2022.findings-acl.222.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization." In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp (2022). "Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 8086–8098. DOI: 10.18653/v1/2022.acl-long.556. URL: https://doi.org/10.18653/v1/2022.acl-long.556.

Luce, R. Duncan (1959). *Individual choice behaviour*. John Wiley.

Lukasik, Michal, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar (2020). "Does label smoothing mitigate label noise?" In: *International Conference on Machine Learning*. PMLR, pp. 6448–6458.

Luo, Bingfeng, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao (2017). "Learning with Noise: Enhance Distantly Supervised Relation Extraction with Dynamic Transition Matrix." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. Ed. by Regina Barzilay and Min-Yen Kan. Association for Computational Linguistics, pp. 430–439. DOI: 10.18653/v1/P17-1040. URL: https://doi.org/10.18653/v1/P17-1040.

Lyu, Yueming and Ivor W. Tsang (2020). "Curriculum Loss: Robust Learning and Generalization against Label Corruption." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rkgt0REKwS.

Ma, Xingjun, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey (July 2018). "Dimensionality-Driven Learning with Noisy Labels." In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 3355–3364. URL: https://proceedings.mlr.press/v80/ma18d.html.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts (June 2011). "Learning Word Vectors for Sentiment Analysis." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. HLT '11. Portland, Oregon: Association for Computational Linguistics, 142–150. ISBN: 9781932432879. URL: https://aclanthology.org/P11-1015/.

Maaten, Laurens Van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11.

Maillard, Jean, Cynthia Gao, Elahe Kalbassi, Kaushik Ram Sadagopan, Vedanuj Goswami, Philipp Koehn, Angela Fan, and Francisco Guzman (July 2023). "Small Data, Big Impact: Leveraging Minimal Data for Effective Machine Translation." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 2740–2756. DOI: 10.18653/v1/2023.acl-long.154. URL: https://aclanthology.org/2023.acl-long.154.

Malach, Eran and Shai Shalev-Shwartz (2017). "Decoupling "when to update" from "how to update"." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information*

Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 960–970. URL: https://proceedings.neurips.cc/paper/2017/hash/58d4d1e7b1e97b258c9ed0b37e02d087-Abstract.html.

Mandal, Devraj, Shrisha Bharadwaj, and Soma Biswas (2020). "A Novel Self-Supervised Re-labeling Approach for Training with Noisy Labels." In: *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*. IEEE, pp. 1370–1379. DOI: 10.1109/WACV45572.2020.9093342. URL: https://doi.org/10.1109/WACV45572.2020.9093342.

Mao, Zhuoyuan and Yen Yu (2024). "Tuning LLMs with Contrastive Alignment Instructions for Machine Translation in Unseen, Low-resource Languages." In: *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*. DOI: 10.18653/v1/2024.loresmt-1.1.

Maystre, Lucas and Matthias Grossglauser (2015). "Fast and Accurate Inference of Plackett-Luce Models." In: *Advances in Neural Information Processing Systems*. Vol. 28.

Menon, Aditya Krishna, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar (2020). "Can gradient clipping mitigate label noise?" In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=rklB76EKPr.

Menon, Aditya Krishna, Brendan van Rooyen, and Nagarajan Natarajan (2016). "Learning from Binary Labels with Instance-Dependent Corruption." In: *CoRR* abs/1605.00751. arXiv: 1605.00751. URL: http://arxiv.org/abs/1605.00751.

Merdjanovska, Elena, Ansar Aynetdinov, and Alan Akbik (Nov. 2024). "NoiseBench: Benchmarking the Impact of Real Label Noise on Named Entity Recognition." In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 18182–18198. DOI: 10.18653/v1/2024.emnlp-main.1011. URL: https://aclanthology.org/2024.emnlp-main.1011.

Ming, Lingfeng, Bo Zeng, Chenyang Lyu, Tianqi Shi, Yu Zhao, Xue Yang, Yefeng Liu, Yiyu Wang, Linlong Xu, Yangyang Liu, et al. (2024). "Marco-LLM: Bridging Languages via Massive Multilingual Training for Cross-Lingual Enhancement." In: *arXiv preprint arXiv:2412.04003*.

Mintz, Mike, Steven Bills, Rion Snow, and Daniel Jurafsky (Aug. 2009). "Distant supervision for relation extraction without labeled data." In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Compu-

tational Linguistics, pp. 1003–1011. URL: https://www.aclweb.org/anthology/P09-1113.

Mishra, Swaroop, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi (May 2022). "Cross-Task Generalization via Natural Language Crowdsourcing Instructions." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 3470–3487. DOI: 10.18653/v1/2022.acl-long.244. URL: https://aclanthology.org/2022.acl-long.244.

Mitchell, Eric, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning (2024). "An Emulator for Fine-tuning Large Language Models using Small Language Models." In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Eo7kv0sllr.

Miyato, Takeru, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii (2018). "Virtual adversarial training: a regularization method for supervised and semi-supervised learning." In: *IEEE transactions on pattern analysis and machine intelligence* 41.8, pp. 1979–1993. URL: https://arxiv.org/abs/1704.03976.

Mosbach, Marius, Maksym Andriushchenko, and Dietrich Klakow (2021). "On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=nzpLWnVAyah.

Moslem, Yasmin, Rejwanul Haque, John D. Kelleher, and Andy Way (June 2023). "Adaptive Machine Translation with Large Language Models." In: *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*. Ed. by Mary Nurminen et al. Tampere, Finland: European Association for Machine Translation, pp. 227–237. URL: https://aclanthology.org/2023.eamt-1.22.

Mosteller, Frederick (1951). "Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations." In: *Psychometrika* 16.1, pp. 3–9. ISSN: 1860-0980. DOI: 10.1007/BF02313422. URL: https://doi.org/10.1007/BF02313422.

Mu, Yongyu, Abudurexiti Reheman, Zhiquan Cao, Yuchun Fan, Bei Li, Yinqiao Li, Tong Xiao, Chunliang Zhang, and Jingbo Zhu (July 2023). "Augmenting Large Language Model Translators via Translation Memories." In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 10287–10299. DOI: 10.18653/v1/2023.findings-acl.653. URL: https://aclanthology.org/2023.findings-acl.653.

Muennighoff, Niklas et al. (2023). "Crosslingual Generalization through Multitask Finetuning." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Ed. by Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki. Association for Computational Linguistics, pp. 15991–16111. DOI: 10.18653/V1/2023.ACL-LONG.891. URL: https://doi.org/10.18653/v1/2023.acl-long.891.

Mukherjee, Subhabrata and Ahmed Hassan Awadallah (2020). "Uncertainty-aware Self-training for Few-shot Text Classification." In: *Advances in Neural Information Processing Systems (NeurIPS 2020)*. Online.

Natarajan, Nagarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari (2013). "Learning with Noisy Labels." In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf.

Nguyen, Duc Tam, Chaithanya Kumar Mummadi, Thi-Phuong-Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox (2020). "SELF: Learning to Filter Noisy Labels with Self-Ensembling." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=HkgsPhNYPS.

Nguyen, Duc Tam, Thi-Phuong-Nhung Ngo, Zhongyu Lou, Michael Klar, Laura Beggel, and Thomas Brox (2019). "Robust Learning Under Label Noise With Iterative Noise-Filtering." In: *CoRR*. arXiv: 1906.00216. URL: http://arxiv.org/abs/1906.00216.

Northcutt, Curtis G., Tailin Wu, and Isaac L. Chuang (2017). "Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels." In: *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. Ed. by Gal Elidan, Kristian Kersting, and Alexander Ihler. AUAI Press. URL: http://auai.org/uai2017/proceedings/papers/35.pdf.

Oliver, Avital, Augustus Odena, Colin Raffel, Ekin Dogus Cubuk, and Ian J. Goodfellow (2018). "Realistic Evaluation of Deep Semi-Supervised Learning Algorithms." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 3239–3250. URL: https://proceedings.neurips.cc/paper/2018/hash/c1fea270c48e8079d8ddf7d06d26ab52-Abstract.html.

OpenAI (2023a). *ChatGPT: Mar 14 Version*. Large language model. URL: https://chat.openai.com/chat (visited on 01/11/2025).

– (2023b). "GPT-4 Technical Report." In: *CoRR* abs/2303.08774. DOI: 10.48550/ARXIV.2303.08774. arXiv: 2303.08774. URL: https://doi.org/10.48550/arXiv.2303.08774.

Ott, Myle, Michael Auli, David Grangier, and Marc'Aurelio Ranzato (2018). "Analyzing Uncertainty in Neural Machine Translation." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 3953–3962. URL: http://proceedings.mlr.press/v80/ott18a.html.

Ouyang, Long et al. (2022). "Training language models to follow instructions with human feedback." In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (July 2002). "Bleu: a Method for Automatic Evaluation of Machine Translation." In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

Patrini, Giorgio, Frank Nielsen, Richard Nock, and Marcello Carioni (2016). "Loss factorization, weakly supervised learning and label noise robustness." In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 708–717. URL: https://proceedings.mlr.press/v48/patrini16.html.

Patrini, Giorgio, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu (2017). "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 2233–2241. DOI: 10.1109/CVPR.2017.240. URL: https://doi.org/10.1109/CVPR.2017.240.

Paul, Debjit, Mittul Singh, Michael A. Hedderich, and Dietrich Klakow (2019). "Handling Noisy Labels for Robustly Learning from Self-Training Data for Low-Resource Sequence Labeling." In: *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 3-5, 2019, Student Research Workshop*. Ed. by Sudipta Kar, Farah Nadeem, Laura Burdick, Greg Durrett, and Na-Rae Han. Association for Computational

Linguistics, pp. 29–34. DOI: 10.18653/v1/n19-3005. URL: https://doi.org/10.18653/v1/n19-3005.

Perez, Ethan, Douwe Kiela, and Kyunghyun Cho (2021). "True Few-Shot Learning with Language Models." In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, pp. 11054–11070. URL: https://proceedings.neurips.cc/paper/2021/hash/5c04925674920eb58467fb52ce4ef728-Abstract.html.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith (Aug. 2019). "To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks." In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, pp. 7–14. DOI: 10.18653/v1/W19-4302. URL: https://aclanthology.org/W19-4302.

Pfeiffer, Jonas, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych (2020). "AdapterHub: A Framework for Adapting Transformers." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54. URL: https://aclanthology.org/2020.emnlp-demos.7/.

Pham, Hieu, Zihang Dai, Qizhe Xie, and Quoc V. Le (2021). "Meta Pseudo Labels." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 11557–11568.

Plackett, Robin L (1975). "The analysis of permutations." In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 24.2, pp. 193–202.

Post, Matt (Oct. 2018). "A Call for Clarity in Reporting BLEU Scores." In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Ed. by Ondřej Bojar et al. Brussels, Belgium: Association for Computational Linguistics, pp. 186–191. DOI: 10.18653/v1/W18-6319. URL: https://aclanthology.org/W18-6319.

Pradhan, Sameer, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong (Aug. 2013). "Towards Robust Linguistic Analysis using OntoNotes." In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 143–152. URL: https://aclanthology.org/W13-3516.

Qin, Yujia et al. (2023). *ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs*. arXiv: 2307.16789 [cs.AI]. URL: https://arxiv.org/abs/2307.16789.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving Language Understanding by Generative Pre-

Training." In: Preprint. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. (2019). *Language models are unsupervised multitask learners*. OpenAI blog.

Rae, Jack W. et al. (2021). "Scaling Language Models: Methods, Analysis & Insights from Training Gopher." In: *CoRR* abs/2112.11446. arXiv: 2112.11446. URL: https://arxiv.org/abs/2112.11446.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn (2023). "Direct Preference Optimization: Your Language Model is Secretly a Reward Model." In: *CoRR* abs/2305.18290. DOI: 10.48550/ARXIV.2305.18290. arXiv: 2305.18290. URL: https://doi.org/10.48550/arXiv.2305.18290.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." In: *J. Mach. Learn. Res.* 21, 140:1–140:67. URL: https://jmlr.org/papers/v21/20-074.html.

Ratner, Alexander, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré (Nov. 2017). "Snorkel: Rapid Training Data Creation with Weak Supervision." In: *Proc. VLDB Endow.* 11.3, 269–282. ISSN: 2150-8097. DOI: 10.14778/3157794.3157797. URL: https://doi.org/10.14778/3157794.3157797.

Raunak, Vikas, Amr Sharaf, Hany Hassan Awadallah, and Arul Menezes (2023). "Leveraging GPT-4 for Automatic Translation Post-Editing." In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. URL: https://aclanthology.org/2023.findings-emnlp.804/.

Reed, Scott E., Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich (2015). "Training Deep Neural Networks on Noisy Labels with Bootstrapping." In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6596.

Rei, Ricardo, José G. C. de Souza, Duarte M. Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luísa Coheur, and André F. T. Martins (2022). "COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task." In: *Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022*. Ed. by Philipp Koehn et al. Association for Computational Linguistics, pp. 578–585. URL: https://aclanthology.org/2022.wmt-1.52.

Rei, Ricardo, Craig Stewart, Ana C Farinha, and Alon Lavie (Nov. 2020). "COMET: A Neural Framework for MT Evaluation." In: *Proceedings*

of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Ed. by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, pp. 2685–2702. DOI: 10.18653/v1/2020.emnlp-main.213. URL: https://aclanthology.org/2020.emnlp-main.213.

Reiss, Frederick, Hong Xu, Bryan Cutler, Karthik Muthuraman, and Zachary Eichenberger (Nov. 2020). "Identifying Incorrect Labels in the CoNLL-2003 Corpus." In: *Proceedings of the 24th Conference on Computational Natural Language Learning*. Ed. by Raquel Fernández and Tal Linzen. Online: Association for Computational Linguistics, pp. 215–226. DOI: 10.18653/v1/2020.conll-1.16. URL: https://aclanthology.org/2020.conll-1.16.

Ren, Mengye, Wenyuan Zeng, Bin Yang, and Raquel Urtasun (2018). "Learning to Reweight Examples for Robust Deep Learning." In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 4331–4340. URL: http://proceedings.mlr.press/v80/ren18a.html.

Ren, Wendi, Yinghao Li, Hanting Su, David Kartchner, Cassie Mitchell, and Chao Zhang (Nov. 2020). "Denoising Multi-Source Weak Supervision for Neural Text Classification." In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Vol. EMNLP 2020. Findings of ACL. Online: Association for Computational Linguistics, pp. 3739–3754. DOI: 10.18653/v1/2020.findings-emnlp.334. URL: https://aclanthology.org/2020.findings-emnlp.334.

Robinson, Joshua David, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka (2021). "Contrastive Learning with Hard Negative Samples." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=CR1XOQ0UTh-.

Rodrigues, Filipe and Francisco C. Pereira (2018). "Deep Learning from Crowds." In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 1611–1618. DOI: 10.1609/AAAI.V32I1.11506. URL: https://doi.org/10.1609/aaai.v32i1.11506.

Rolnick, David, Andreas Veit, Serge J. Belongie, and Nir Shavit (2017). "Deep Learning is Robust to Massive Label Noise." In: *CoRR*. arXiv: 1705.10694. URL: http://arxiv.org/abs/1705.10694.

Rooyen, Brendan van, Aditya Krishna Menon, and Robert C. Williamson (2015). "Learning with Symmetric Label Noise: The Importance of

Being Unhinged." In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, pp. 10–18. URL: https://proceedings.neurips.cc/paper/2015/hash/45c48cce2e2d7fbdea1afc51c7c6ad26-Abstract.html.

Rücker, Susanna and Alan Akbik (Dec. 2023). "CleanCoNLL: A Nearly Noise-Free Named Entity Recognition Dataset." In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 8628–8645. DOI: 10.18653/v1/2023.emnlp-main.533. URL: https://aclanthology.org/2023.emnlp-main.533.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." In: *CoRR* abs/1910.01108. arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

Sanh, Victor, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. (2022). "Multitask Prompted Training Enables Zero-Shot Task Generalization." In: *International Conference on Learning Representations*. URL: https://arxiv.org/abs/2110.08207.

Sarti, Gabriele, Phu Mon Htut, Xing Niu, Benjamin Hsu, Anna Currey, Georgiana Dinu, and Maria Nadejde (July 2023). "RAMP: Retrieval and Attribute-Marking Enhanced Prompting for Attribute-Controlled Translation." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 1476–1490. DOI: 10.18653/v1/2023.acl-short.126. URL: https://aclanthology.org/2023.acl-short.126.

Scao, Teven Le et al. (2022). "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model." In: *CoRR* abs/2211.05100. DOI: 10.48550/ARXIV.2211.05100. arXiv: 2211.05100. URL: https://doi.org/10.48550/arXiv.2211.05100.

Schick, Timo and Hinrich Schütze (2022). "True Few-Shot Learning with Prompts—A Real-World Perspective." In: *Transactions of the Association for Computational Linguistics* 10, pp. 716–731. DOI: 10.1162/tacl_a_00485. URL: https://aclanthology.org/2022.tacl-1.41.

Schmidt, Fabian David, Ivan Vulić, and Goran Glavaš (Dec. 2022). "Don't Stop Fine-Tuning: On Training Regimes for Few-Shot Cross-Lingual Transfer with Multilingual Language Models." In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language*

Processing. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 10725–10742. DOI: 10.18653/v1/2022.emnlp-main.736. URL: https://aclanthology.org/2022.emnlp-main.736.

Schmidt, Fabian David, Ivan Vulić, and Goran Glavaš (2023). "Free Lunch: Robust Cross-Lingual Transfer via Model Checkpoint Averaging." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 5712–5730. DOI: 10.18653/v1/2023.acl-long.314. URL: https://aclanthology.org/2023.acl-long.314.

Schroff, Florian, Antonio Criminisi, and Andrew Zisserman (2011). "Harvesting Image Databases from the Web." In: *IEEE Trans. Pattern Anal. Mach. Intell.* 33.4, pp. 754–766. DOI: 10.1109/TPAMI.2010.133. URL: https://doi.org/10.1109/TPAMI.2010.133.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal Policy Optimization Algorithms." In: *CoRR* abs/1707.06347. arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "Improving Neural Machine Translation Models with Monolingual Data." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, pp. 86–96. DOI: 10.18653/v1/P16-1009. URL: https://aclanthology.org/P16-1009.

Shaham, Uri, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal (2024). "Multilingual Instruction Tuning With Just a Pinch of Multilinguality." In: *Findings of the Association for Computational Linguistics ACL 2024*. DOI: 10.18653/v1/2024.findings-acl.136.

Shu, Jun, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng (2019). "Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting." 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*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 1917–1928. URL: https://proceedings.neurips.cc/paper/2019/hash/e58cc5ca94270acaceed13bc82dfedf7-Abstract.html.

Sia, Suzanna, David Mueller, and Kevin Duh (2024). "Where does In-context Translation Happen in Large Language Models." In: *arXiv preprint*. URL: https://arxiv.org/abs/2403.04510.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search." In: *Nat.* 529.7587, pp. 484–489. DOI: 10.1038/NATURE16961. URL: https://doi.org/10.1038/nature16961.

Smith, Ryan, Jason A. Fries, Braden Hancock, and Stephen H. Bach (2022). "Language Models in the Loop: Incorporating Prompting into Weak Supervision." In: *CoRR* abs/2205.02318. DOI: 10.48550/arXiv.2205.02318. arXiv: 2205.02318. URL: https://doi.org/10.48550/arXiv.2205.02318.

Song, Feifan, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang (2023). "Preference Ranking Optimization for Human Alignment." In: *CoRR* abs/2306.17492. DOI: 10.48550/ARXIV.2306.17492. arXiv: 2306.17492. URL: https://doi.org/10.48550/arXiv.2306.17492.

Song, Hwanjun, Minseok Kim, and Jae-Gil Lee (2019). "SELFIE: Refurbishing Unclean Samples for Robust Deep Learning." In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5907–5915. URL: https://proceedings.mlr.press/v97/song19b.html.

Song, Hwanjun, Minseok Kim, Dongmin Park, and Jae-Gil Lee (2019). "Prestopping: How Does Early Stopping Help Generalization against Label Noise?" In: *CoRR* abs/1911.08059. arXiv: 1911.08059. URL: http://arxiv.org/abs/1911.08059.

Song, Hwanjun, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee (2022). "Learning from Noisy Labels with Deep Neural Networks: A Survey." In: *IEEE Transactions on Neural Networks and Learning Systems*.

Srivastava, Saurabh, Chengyue Huang, Weiguo Fan, and Ziyu Yao (2023). "Instance Needs More Care: Rewriting Prompts for Instances Yields Better Zero-Shot Performance." In: *CoRR* abs/2310.02107. DOI: 10.48550/ARXIV.2310.02107. arXiv: 2310.02107. URL: https://doi.org/10.48550/arXiv.2310.02107.

Stap, David, Eva Hasler, Bill Byrne, Christof Monz, and Ke Tran (2024). "The Fine-Tuning Paradox: Boosting Translation Quality Without Sacrificing LLM Abilities." In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. DOI: 10.18653/v1/2024.acl-long.336. URL: https://aclanthology.org/2024.acl-long.336.

Stephan, Andreas, Vasiliki Kougia, and Benjamin Roth (Dec. 2022). "SepLL: Separating Latent Class Labels from Weak Supervision Noise." In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 3918–3929. URL: https://aclanthology.org/2022.findings-emnlp.288.

Su, Jianlin, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu (2024). "RoFormer: Enhanced transformer with

Rotary Position Embedding." In: *Neurocomputing* 568, p. 127063. DOI: 10.1016/J.NEUCOM.2023.127063. URL: https://doi.org/10.1016/j.neucom.2023.127063.

Sukhbaatar, Sainbayar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus (2015). "Training convolutional networks with noisy labels." In: *3rd International Conference on Learning Representations, ICLR 2015*.

Sun, Yi, Yan Tian, Yiping Xu, and Jianxiang Li (2019). "Limited Gradient Descent: Learning With Noisy Labels." In: *IEEE Access* 7, pp. 168296–168306. DOI: 10.1109/ACCESS.2019.2954547. URL: https://doi.org/10.1109/ACCESS.2019.2954547.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna (2016). "Rethinking the Inception Architecture for Computer Vision." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.

Tan, Cheng, Jun Xia, Lirong Wu, and Stan Z. Li (2021). "Co-learning: Learning from Noisy Labels with Self-supervision." In: *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*. Ed. by Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran. ACM, pp. 1405–1413. DOI: 10.1145/3474085.3475622. URL: https://doi.org/10.1145/3474085.3475622.

Tanaka, Daiki, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa (2018). "Joint Optimization Framework for Learning With Noisy Labels." In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 5552–5560. DOI: 10.1109/CVPR.2018.00582. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Tanaka_Joint_Optimization_Framework_CVPR_2018_paper.html.

Tang, Yixuan and Yi Yang (2024). *MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries*. arXiv: 2401.15391 [cs.CL]. URL: https://arxiv.org/abs/2401.15391.

Tanno, Ryutaro, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman (2019). "Learning From Noisy Labels by Regularized Estimation of Annotator Confusion." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 11244–11253. DOI: 10.1109/CVPR.2019.01150.

Tänzer, Michael, Sebastian Ruder, and Marek Rei (2021). "BERT memorisation and pitfalls in low-resource scenarios." In: *CoRR* abs/2105.00828. arXiv: 2105.00828. URL: https://arxiv.org/abs/2105.00828.

– (2022). "Memorisation versus Generalisation in Pre-trained Language Models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022,*

Dublin, Ireland, May 22-27, 2022. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 7564–7578. DOI: 10.18653/v1/2022.acl-long.521. URL: https://doi.org/10.18653/v1/2022.acl-long.521.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto (2023). *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. URL: https://github.com/tatsu-lab/stanford_alpaca.

Tarvainen, Antti and Harri Valpola (2017). "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=ry8u21rtl.

Thurstone, Louis L (1927). "Psychophysical analysis." In: *The American journal of psychology* 38.3, pp. 368–389.

Tiedemann, Jörg and Santhosh Thottingal (Nov. 2020). "OPUS-MT – Building open translation services for the World." In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Ed. by André Martins et al. Lisboa, Portugal: European Association for Machine Translation, pp. 479–480. URL: https://aclanthology.org/2020.eamt-1.61.

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147. URL: https://aclanthology.org/W03-0419.

Touvron, Hugo et al. (2023a). "LLaMA: Open and Efficient Foundation Language Models." In: *CoRR* abs/2302.13971. DOI: 10.48550/ARXIV.2302.13971. arXiv: 2302.13971. URL: https://doi.org/10.48550/arXiv.2302.13971.

Touvron, Hugo et al. (2023b). "Llama 2: Open Foundation and Fine-Tuned Chat Models." In: *CoRR* abs/2307.09288. DOI: 10.48550/ARXIV.2307.09288. arXiv: 2307.09288. URL: https://doi.org/10.48550/arXiv.2307.09288.

Train, Kenneth (2003). *Discrete Choice Method With Simulation*. Cambridge University Press. DOI: 10.1017/CBO9780511753930.

Üstün, Ahmet, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. (2024). "Aya model: An instruction finetuned open-access multilingual language model." In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. DOI: 10.18653/v1/2024.acl-long.845.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017).

"Attention is All you Need." In: *Advances in Neural Information Processing Systems*. URL: https://arxiv.org/abs/1706.03762.

Veit, Andreas, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie (2017). "Learning from Noisy Large-Scale Datasets with Minimal Supervision." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, pp. 6575–6583. DOI: 10.1109/CVPR.2017.696. URL: https://doi.org/10.1109/CVPR.2017.696.

Vilar, David, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster (July 2023). "Prompting PaLM for Translation: Assessing Strategies and Performance." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 15406–15427. DOI: 10.18653/v1/2023.acl-long.859. URL: https://aclanthology.org/2023.acl-long.859.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman (2019a). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." In: In the Proceedings of ICLR.

Wang, Hao, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li (2019b). "Learning with Noisy Labels for Sentence-level Sentiment Classification." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, pp. 6285–6291. DOI: 10.18653/V1/D19-1655. URL: https://doi.org/10.18653/v1/D19-1655.

Wang, Yaqing, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah (2020). "Adaptive self-training for few-shot neural sequence labeling." In: *arXiv preprint arXiv:2010.03680*.

Wang, Yisen, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey (2019c). "Symmetric Cross Entropy for Robust Learning With Noisy Labels." In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, pp. 322–330. DOI: 10.1109/ICCV.2019.00041. URL: https://doi.org/10.1109/ICCV.2019.00041.

Wang, Yizhong, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi (2023). *Self-Instruct: Aligning Language Models with Self-Generated Instructions*. arXiv: 2212.10560 [cs.CL]. URL: https://arxiv.org/abs/2212.10560.

Wang, Zihan, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han (Nov. 2019d). "CrossWeigh: Training Named Entity Tagger from Imperfect Annotations." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 5154–5163. DOI: 10.18653/v1/D19-1519. URL: https://aclanthology.org/D19-1519.

Warstadt, Alex, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman (Nov. 2020). "Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually)." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 217–235. DOI: 10.18653/v1/2020.emnlp-main.16. URL: https://aclanthology.org/2020.emnlp-main.16.

Wei, Hongxin, Lei Feng, Xiangyu Chen, and Bo An (2020). "Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, pp. 13723–13732. DOI: 10.1109/CVPR42600.2020.01374. URL: https://doi.org/10.1109/CVPR42600.2020.01374.

Wei, Jason, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le (2022a). "Finetuned Language Models are Zero-Shot Learners." In: *International Conference on Learning Representations*. URL: https://arxiv.org/abs/2109.01652.

Wei, Jason, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. (2022b). "Emergent abilities of large language models." In: *arXiv preprint arXiv:2206.07682*.

Wolf, Thomas et al. (Oct. 2020). "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Di, Shaomu Tan, Yan Meng, David Stap, and Christof Monz (2024a). "How Far can 100 Samples Go? Unlocking Zero-Shot Translation with Tiny Multi-Parallel Data." In: *Findings of the Association for Computational Linguistics ACL 2024*. URL: https://aclanthology.org/2024.findings-acl.896.

Wu, Minghao, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari (2024b). "Adapting Large Language Models for Document-Level Machine Translation." In: *arXiv preprint*. URL: https://arxiv.org/abs/2401.06468.

Wu, Minghao, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Aji (2024c). "LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions." In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*. URL: https://aclanthology.org/2024.eacl-long.57.

Wu, Shijie and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." 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)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077.

Wu, Tianhao, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar (2024d). *Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge*. arXiv: 2407.19594 [cs.CL]. URL: https://arxiv.org/abs/2407.19594.

Xia, Xiaobo, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang (2021). "Robust early-learning: Hindering the memorization of noisy labels." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=Eql5b1_hTE4.

Xia, Yan, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun (2015). "Learning Discriminative Reconstructions for Unsupervised Outlier Removal." In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, pp. 1511–1519. DOI: 10.1109/ICCV.2015.177. URL: https://doi.org/10.1109/ICCV.2015.177.

Xiao, Tong, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang (2015). "Learning from massive noisy labeled data for image classification." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, pp. 2691–2699. DOI: 10.1109/CVPR.2015.7298885. URL: https://doi.org/10.1109/CVPR.2015.7298885.

Xie, Qizhe, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le (2020a). "Unsupervised Data Augmentation for Consistency Training." In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html.

Xie, Qizhe, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le (2020b). "Self-Training With Noisy Student Improves ImageNet Classification." In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 10684–10695. DOI: 10.1109/CVPR42600.2020.01070.

Xu, Haoran, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla (2023). "A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models." In: *CoRR* abs/2309.11674. DOI: 10.48550/ARXIV.2309.11674. arXiv: 2309.11674. URL: https://doi.org/10.48550/arXiv.2309.11674.

– (2024a). "A Paradigm Shift in Machine Translation: Boosting Translation Performance of Large Language Models." In: *The Twelfth International Conference on Learning Representations*. URL: https://openreview.net/forum?id=farT6XXntP.

Xu, Haoran, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim (2024b). "Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation." In: *Proceedings of the 41st International Conference on Machine Learning*. URL: https://proceedings.mlr.press/v235/xu24t.html.

Yan, Yan, Romer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy (2010). "Modeling annotator expertise: Learning when everybody knows a bit of something." In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, pp. 932–939. URL: https://proceedings.mlr.press/v9/yan10a.html.

Yan, Yan, Zhongwen Xu, Ivor W. Tsang, Guodong Long, and Yi Yang (2016). "Robust Semi-Supervised Learning through Label Aggregation." In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. Ed. by Dale Schuurmans and Michael P. Wellman. AAAI Press, pp. 2244–2250. DOI: 10.1609/AAAI.V30I1.10276. URL: https://doi.org/10.1609/aaai.v30i1.10276.

Yao, Yazhou, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang (2021). "Jo-SRC: A Contrastive Approach for Combating Noisy Labels." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 5192–5201. DOI: 10.1109/CVPR46437.2021.00515.

Yao, Yu, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama (2020). "Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning." In: *Advances in Neural Information Processing Systems 33: Annual Con-*

ference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin.

Yarowsky, David (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods." In: *33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings*. Ed. by Hans Uszkoreit. Morgan Kaufmann Publishers / ACL, pp. 189–196. DOI: 10.3115/981658.981684. URL: https://aclanthology.org/P95-1026/.

Yi, Kun and Jianxin Wu (2019). "Probabilistic End-To-End Noise Correction for Learning With Noisy Labels." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, pp. 7017–7025. DOI: 10.1109/CVPR.2019.00718.

Yu, Xingrui, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama (2019). "How does Disagreement Help Generalization against Label Corruption?" In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 7164–7173. URL: http://proceedings.mlr.press/v97/yu19b.html.

Yu, Yue, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang (2021). "Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach." In: *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*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Association for Computational Linguistics, pp. 1063–1077. DOI: 10.18653/v1/2021.naacl-main.84. URL: https://doi.org/10.18653/v1/2021.naacl-main.84.

Yuan, Zheng, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang (2023). "RRHF: Rank Responses to Align Language Models with Human Feedback without tears." In: *CoRR* abs/2304.05302. DOI: 10.48550/ARXIV.2304.05302. arXiv: 2304.05302. URL: https://doi.org/10.48550/arXiv.2304.05302.

Zaken, Elad Ben, Yoav Goldberg, and Shauli Ravfogel (2022). "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 1–9. DOI: 10.18653/v1/2022.acl-short.1. URL: https://doi.org/10.18653/v1/2022.acl-short.1.

Zeng, Jiali, Fandong Meng, Yongjing Yin, and Jie Zhou (2023). "Tim: Teaching large language models to translate with comparison." In: *arXiv preprint arXiv:2307.04408*.

– (2024). "Teaching Large Language Models to Translate with Comparison." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. URL: https://ojs.aaai.org/index.php/AAAI/article/view/29920.

Zhang, Biao, Barry Haddow, and Alexandra Birch (2023). "Prompting Large Language Model for Machine Translation: A Case Study." In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 41092–41110. URL: https://proceedings.mlr.press/v202/zhang23m.html.

Zhang, Biao, Zhongtao Liu, Colin Cherry, and Orhan Firat (2024a). "When Scaling Meets LLM Finetuning: The Effect of Data, Model and Finetuning Method." In: *The Twelfth International Conference on Learning Representations*. URL: https://arxiv.org/abs/2402.17193.

Zhang, Bowen, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki (2021a). "FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling." In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, pp. 18408–18419.

Zhang, Chang-Bin, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng (2021b). "Delving Deep Into Label Smoothing." In: *IEEE Trans. Image Process.* 30, pp. 5984–5996. DOI: 10.1109/TIP.2021.3089942. URL: https://doi.org/10.1109/TIP.2021.3089942.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (2017). "Understanding deep learning requires rethinking generalization." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=Sy8gdB9xx.

Zhang, Jiajie et al. (2024b). *LongCite: Enabling LLMs to Generate Fine-grained Citations in Long-context QA*. arXiv: 2409.02897 [cs.CL]. URL: https://arxiv.org/abs/2409.02897.

Zhang, Jieyu, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner (2022). "A Survey on Programmatic Weak Supervision." In: *CoRR* abs/2202.05433. arXiv: 2202.05433. URL: https://arxiv.org/abs/2202.05433.

Zhang, Jieyu, Yue Yu, NameError, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner (2021c). "WRENCH: A Comprehensive

Benchmark for Weak Supervision." In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*. Ed. by Joaquin Vanschoren and Sai-Kit Yeung.

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter J. Liu (2020). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 11328–11339. URL: http://proceedings.mlr.press/v119/zhang20ae.html.

Zhang, Miaoran, Vagrant Gautam, Mingyang Wang, Jesujoba O. Alabi, Xiaoyu Shen, Dietrich Klakow, and Marius Mosbach (2024c). *The Impact of Demonstrations on Multilingual In-Context Learning: A Multidimensional Analysis*. arXiv: 2402.12976 [cs.CL].

Zhang, Shaolei, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, et al. (2023a). "BayLing: Bridging Cross-lingual Alignment and Instruction Following through Interactive Translation for Large Language Models." In: *arXiv preprint*. URL: https://arxiv.org/abs/2306.10968.

Zhang, Xiang, Junbo Jake Zhao, and Yann LeCun (2015a). "Character-level Convolutional Networks for Text Classification." In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, pp. 649–657.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015b). "Character-level convolutional networks for text classification." In: *Advances in neural information processing systems* 28.

Zhang, Xuan, Navid Rajabi, Kevin Duh, and Philipp Koehn (Dec. 2023b). "Machine Translation with Large Language Models: Prompting, Few-shot Learning, and Fine-tuning with QLoRA." In: *Proceedings of the Eighth Conference on Machine Translation*. Singapore: Association for Computational Linguistics, pp. 466–479. DOI: 10.18653/v1/2023.wmt-1.43. URL: https://aclanthology.org/2023.wmt-1.43.

Zhang, Yikai, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen (2021d). "Learning with Feature-Dependent Label Noise: A Progressive Approach." In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=ZPa2SyGcbwh.

Zhang, Zhilu and Mert R. Sabuncu (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. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 8792–8802.

Zhao, Wayne Xin et al. (2024). *A Survey of Large Language Models*. arXiv: 2303.18223 [cs.CL]. URL: https://arxiv.org/abs/2303.18223.

Zhao, Zihao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh (2021). "Calibrate Before Use: Improving Few-shot Performance of Language Models." In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 12697–12706. URL: http://proceedings.mlr.press/v139/zhao21c.html.

Zheng, Guoqing, Ahmed Hassan Awadallah, and Susan T. Dumais (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*. AAAI Press, pp. 11053–11061. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17319.

Zheng, Guoqing, Giannis Karamanolakis, Kai Shu, and Ahmed Awadallah (July 2022a). "WALNUT: A Benchmark on Semi-weakly Supervised Learning for Natural Language Understanding." In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 873–899. DOI: 10.18653/v1/2022.naacl-main.64. URL: https://aclanthology.org/2022.naacl-main.64.

Zheng, Lianmin et al. (2023). "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena." In: *CoRR* abs/2306.05685. DOI: 10.48550/ARXIV.2306.05685. arXiv: 2306.05685. URL: https://doi.org/10.48550/arXiv.2306.05685.

Zheng, Songzhu, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris N. Metaxas, and Chao Chen (2020). "Error-Bounded Correction of Noisy Labels." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 11447–11457. URL: http://proceedings.mlr.press/v119/zheng20c.html.

Zheng, Yanan, Jing Zhou, Yujie Qian, Ming Ding, Chonghua Liao, Li Jian, Ruslan Salakhutdinov, Jie Tang, Sebastian Ruder, and Zhilin Yang (2022b). "FewNLU: Benchmarking State-of-the-Art Methods for Few-Shot Natural Language Understanding." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio.

Association for Computational Linguistics, pp. 501–516. DOI: 10.18653/v1/2022.acl-long.38. URL: https://doi.org/10.18653/v1/2022.acl-long.38.

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. (2023a). "Lima: Less is more for alignment." In: *arXiv preprint arXiv:2305.11206*.

Zhou, Chunting, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. (2023b). "LIMA: Less Is More for Alignment." In: *Thirty-seventh Conference on Neural Information Processing Systems*. URL: https://arxiv.org/abs/2305.11206.

Zhou, Jiawei et al. (2022). "Hyperlink-induced Pre-training for Passage Retrieval in Open-domain Question Answering." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Association for Computational Linguistics, pp. 7135–7146. DOI: 10.18653/v1/2022.acl-long.493. URL: https://doi.org/10.18653/v1/2022.acl-long.493.

Zhou, Tianyi, Shengjie Wang, and Jeff A. Bilmes (2021). "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. URL: https://openreview.net/forum?id=lmTWnm3coJJ.

Zhou, Wangchunshu, Canwen Xu, and Julian McAuley (2022). "BERT Learns to Teach: Knowledge Distillation with Meta Learning." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Zhou, Wenxuan, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren (2020). "NERO: A Neural Rule Grounding Framework for Label-Efficient Relation Extraction." In: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. Ed. by Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen. ACM / IW3C2, pp. 2166–2176. DOI: 10.1145/3366423.3380282. URL: https://doi.org/10.1145/3366423.3380282.

Zhu, Dawei, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow (2022). "Is BERT Robust to Label Noise? A Study on Learning with Noisy Labels in Text Classification." In: *Proceedings of the Third Workshop on Insights from Negative Results in NLP, Insights@ACL 2022, Dublin, Ireland, May 26, 2022*. Ed. by Shabnam Tafreshi, João Sedoc, Anna Rogers, Aleksandr Drozd, Anna Rumshisky, and Arjun R. Akula. Association for Computational Linguistics, pp. 62–67. DOI: 10.18653/v1/2022.insights-1.8. URL: https://doi.org/10.18653/v1/2022.insights-1.8.

Zhu, Dawei, Xiaoyu Shen, Michael A. Hedderich, and Dietrich Klakow (2023a). "Meta Self-Refinement for Robust Learning with Weak

Supervision." In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Association for Computational Linguistics, pp. 1043–1058. URL: https://aclanthology.org/2023.eacl-main.74.

Zhu, Dawei, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow (July 2023b). "Weaker Than You Think: A Critical Look at Weakly Supervised Learning." In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 14229–14253. DOI: 10.18653/v1/2023.acl-long.796. URL: https://aclanthology.org/2023.acl-long.796.

Zhu, Dawei, Sony Trenous, Xiaoyu Shen, Dietrich Klakow, Bill Byrne, and Eva Hasler (2024). "A Preference-driven Paradigm for Enhanced Translation with Large Language Models." In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. DOI: 10.18653/v1/2024.naacl-long.186.

Zhu, Zhaowei, Tongliang Liu, and Yang Liu (2021). "A Second-Order Approach to Learning With Instance-Dependent Label Noise." In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 10113–10123. DOI: 10.1109/CVPR46437.2021.00998.

Zoph, Barret, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le (2020). "Rethinking Pre-training and Self-training." In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. URL: https://proceedings.neurips.cc/paper/2020/hash/27e9661e033a73a6ad8cefcde965c54d-Abstract.html.