FEW-SHOT INTENT DETECTION WITH PRE-TRAINED LANGUAGE MODELS: TRANSFERABILITY, EXPRESSIVENESS AND EFFICIENCY

HAODE ZHANG

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University
Department of Computing

FEW-SHOT INTENT DETECTION WITH PRE-TRAINED LANGUAGE MODELS: TRANSFERABILITY, EXPRESSIVENESS AND EFFICIENCY

Haode Zhang

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
September 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Haode Zhang

Abstract

The identification of user intents is a fundamental component of a task-oriented dialogue system, with the aim of detecting the intent underlying a user's utterance, according to which an appropriate response is provided. Typically, intent detection is formulated into a text classification task, which has benefited from the success of deep learning techniques. However, the acquisition of a large number of annotations for training is expensive. This thesis addresses the challenge of few-shot intent detection, whereby the goal is to develop a highly effective intent classifier using only a limited amount of annotated data, thereby improving data efficiency.

We first study the cross-domain transferability for few-shot intent detection, exploring the possibility of jointly utilizing abundant labeled data in a source domain and easily available unlabeled data in a target domain to train an intent classifier with reasonable performance. We investigate techniques of transfer learning across domains and adapting to a new domain. Leveraging the data in public intent detection datasets, we train IntentBERT, the backbone that transfers knowledge from diverse multiple intent detection domains, significantly improving the performance in the target domain. With easily available unlabeled data in the target domain, the performance is further enhanced.

Next, to improve the expressiveness of IntentBERT, the study focuses on a particular property of the pre-trained language models (PLMs) – anisotropy, an undesirable geometric property of the feature space. We discover that supervised pre-training

yields an anisotropic feature space, which may suppress the expressive power of the semantic representations. To mitigate the problem, we propose to enhance supervised pre-training by regularizing the feature space towards isotropy. We propose two regularizers based on contrastive learning and correlation matrix respectively, and demonstrate their effectiveness through extensive experiments. Through the joint supervised pre-training and isotropization, we achieve improved performance in few-shot intent detection.

Then, to further improve the data efficiency, we revisit the overfitting phenomenon, continual pre-training, and direct fine-tuning based on PLMs in the context of few-shot intent detection. Although the prevailing approach to few-shot intent detection is continual pre-training, i.e., fine-tuning PLMs on external resources, our study demonstrates that continual pre-training may not be necessary. Specifically, we find that the overfitting issue of PLMs may not be as severe as previously believed, i.e. directly fine-tuning PLMs with only a handful of labeled examples already yields decent results, and the performance gap quickly shrinks as the number of labeled data grows. We further enhance the performance of direct fine-tuning with context augmentation and sequential self-distillation. Comprehensive experiments on real-world benchmarks show that given only two or more labeled samples per class, the enhanced direct fine-tuning outperforms many strong baselines that utilize external data sources for continual pre-training.

Finally, to enhance the computational efficiency, we study model compression for intent detection with limited labeled data. Traditional approaches to model compression, such as model pruning and distillation, typically rely on access to large amounts of data. However, such datasets are not readily available under the few-shot scenario. To overcome this challenge, we propose a scheme that capitalizes on off-the-shelf generative PLMs for data augmentation. Furthermore, we introduce a vocabulary pruning technique employing a nearest neighbour matching scheme. Through extensive experiments, we demonstrate the efficacy of the proposed method – we can

compress the model by a factor of 21, and thus enable the deployment of the model in resource-constrained scenarios, including mobile devices and embedded systems.

The results have been published in or submitted to various top natural language processing conferences, including Findings of EMNLP-2021 [119], NAACL-2022 (oral) [118] and Findings of ACL 2023 [117].

Publications Arising from the Thesis

1. Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. a. Effectiveness of pre-training for few-shot intent classification. In Findings of EMNLP 2021, short paper.

2. Haode Zhang, Haowen Liang, Yuwei Zhang, Liming Zhan, Xiao-Ming Wu, Xiaolei Lu, Albert Y.S. La. Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization. In NAACL 2022, long paper, oral.

3. Haode Zhang, Haowen Liang, Liming Zhan, Xiao-Ming Wu, Albert Y.S. Lam. Revisit Few-shot Intent Classification with PLMs: Direct Fine-tuning vs. Continual Pre-training. In Findings of ACL 2023, long paper.

4. Zhang Yuwei, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, Albert Lam. New intent discovery with pre-training and contrastive learning. In ACL 2022, long paper.

Acknowledgments

First, I would like to express my utmost appreciation to my supervisor, Prof. Xiao-Ming Wu, for her invaluable guidance throughout my doctoral journey. Her support, meticulousness, passion, and positive outlook have profoundly influenced me, and I will forever treasure these qualities. Without her mentorship and encouragement, this thesis would hardly be successful completed.

Second, I would like to thank my lab mates for their friendship and support. I would like to thank Yuwei Zhang, Haowen Liang and Liming Zhan for their support.

Finally, I would like to express my gratitude to my family for their continuous and unparalleled love, help, and support. Especially, I would like to thank my sister for her impressive optimism, strong will and sense of responsibility.

I would also thank myself, it is an exceptionally long journey before I come here.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Building dialogue systems with the ability to interact with users in natural languages is a long-standing goal of artificial intelligence. Such systems can be categorized into open dialogue (OD) systems and task-oriented dialogue (TOD) systems according to different objectives – OD systems are designed for general chatting, while TOD systems to assist users handle specific tasks, ranging from financial services and medical consultations to online shopping. Within a TOD system, intent detection is a critical module, as it enables accurate understanding of user intents, and thus facilitates dialogue management, supports task fulfillment, enhances natural language understanding, handles errors, and allows customization and adaptation. It forms a fundamental component in building intelligent conversational agents.

Intent detection is formulated into a text classification task. Contemporary machine learning models, particularly those grounded in deep learning models, have achieved impressive success in the task. Nevertheless, the training of such models usually demands a huge amount of labeled data, which is prohibitive to obtain. Consequently, over the preceding decade, few-shot intent detection, i.e. training a well-performing intent classifier with only a few labeled data, has attracted substantial interests in the community.

To tackle few-shot intent detection, earlier works mainly focus on the design of novel model architectures and training paradigms. These works encompass induction network [32] based on capsule networks, convolutional-neural-network-based models [115], joint intent detection and slot-filling in the meta-learning framework [7] and metric learning [72]. Since the emergence of pre-trained language models (PLMs) [17], the landscape of natural language processing (NLP) has undergone a profound transformation. This paradigm shift has engendered advancements across an extensive spectrum of NLP tasks, including text classification, sequence labeling, machine translation, dialogue systems, and text generation. The power of PLMs lies in the general knowledge learned from huge corpora, which is transferable to specific NLP tasks. Concretely, a deep model is first pre-trained with massive amounts of data, often in an unsupervised manner, thereby endowing the model with a profound understanding of the language, including linguistic architectures, contextual inter-dependencies, and semantic representations. Over the preceding years, PLMs have reshaped the outlook of the NLP field, but it remains a challenge how we can apply PLMs to few-shot intent detection.

First, although PLMs have learned extensive knowledge from huge corpus, it has been shown that the continual pre-training on relevant corpus or tasks benefit the down-stream tasks [36]. Efforts have been dedicated to adapting pre-trained language models to a specific task such as intent detection by conducting continued pre-training, with large unlabeled dialogue corpus [36], natural language inference (NLI) tasks [121] and fake intent detection data generated from wikiHow database [122]. However, these solutions usually take a substantial volume of data for continual pre-training. For instance, [121] adopts 1 million NLI pairs for continual pre-training. These works neglect the existence of cross-domain intent detection data. Consequently, the related issues are to be explored, including the transferability of the knowledge learned from source domains to the semantically irrelevant domain, along with method that considers both domain shift and domain-specific structure.

Second, the expressiveness of PLMs have been known to be constrained by the anisotropy of the feature space. Anisotropy is a geometric property that semantic vectors fall into a narrow cone. It has been identified as a crucial factor for the sub-optimal performance of PLMs on a variety of downstream tasks [30], which is also known as the representation degeneration problem [30]. However, the interaction between the aforementioned pre-training process and isotropization of PLMs is still under explored. Furthermore, novel techniques to perform isotropization during continual pre-training to yield better performance are to be developed. Consequently, we need a closer examination of the isotropy for few-shot intent detection.

Third, current main-stream approaches rely on extra data to learn the transferable knowledge, which enhances the complexity of the pipeline, incurs the overhead of extra computational resources. However, intuitively, as the number of the few data increases, the necessity of the continual pre-training diminishes. Contemporary endeavors focus on the continual pre-training stage, neglecting how to better exploit the few data. Therefore, a comprehensive investigation is needed to unlock the latent potential of the few data at hands, based on which a discussion of the necessity of the continual pre-training should be conducted.

Furthermore, a major limitation of PLMs is the large sizes, usually containing more than one million parameters [78]. Consequently, PLM-based few-shot intent detection solutions usually incur considerable computational overhead, necessitating substantial computational resources, including high-performance processors, memory capacities, and power consumption. Such overheads pose challenges when the solution is to be deployed on resource-constrained devices such as edge devices and mobile devices. However, PLM compression under few-shot scenario receives less attention. Some effective model compression methods such as knowledge distillation do not work well under few-shot scenarios. Therefore, a more comprehensive investigation into model compression for few-shot intent detection is valuable.

In this thesis, we aim to address the issues as mentioned above. We conduct a

comprehensive study of few-shot intent detection based on PLMs, encompassing the dimensions of transferability, expressiveness and efficiency. Specifically, we made the following contributions.

Contribution 1: the study of the cross-domain transferability. We study cross-domain few-shot intent detection and demonstrate the feasibility of transferring knowledge from abundant labeled data in source domains to tackle few-shot intent detection. We also investigate techniques to jointly utilize data from both source domain and target domain, to better adapt the model to the target domain. Extensive experiments on real-world benchmark datasets show consistent improvements of the proposed methods over competitive baselines, demonstrating the cross-domain transferability for few-shot intent detection. Leveraging such transferability, we propose IntentBERT, a backbone for few-shot intent detection, which is obtained by continual pre-training of BERT on labeled utterances from public dataset. IntentBERT features not only performance superiority over various competitive baselines, but also superior data-efficiency – it consumes much less data for continual pre-training. The findings were published in the Findings of EMNLP-2021 [119].

Contribution 2: new isotropization techniques to boost the expressiveness. Following the validation of cross-domain transferability, we proceed to conduct an in-depth analysis of the impact of continual pre-training on the isotropy. It is revealed that supervised pre-training renders the feature space more anisotropic, which suppresses the expressive power. To mitigate the anisotropy, we devise an innovative framework involving joint supervised pre-training and isotropization, wherein two regularizers are introduced to generate an isotropic feature space. The efficacy of the framework is demonstrated through extensive experiments, showing that it is promising to regularize supervised pre-training towards isotropy to enhance the performance of few-shot intent detection. The findings were published in NAACL-2022 [118].

Contribution 3: new technique to better exploit the few data To further enhance the data-efficiency, we examine the prospect of eliminating the stage of con-

tinual pre-training. We commence by the thorough analysis of direct fine-tuning PLMs with only a few data, which is commonly perceived as a bad practice due to severe overfitting. It is found that the process already yields decent performance, compared to methods involving continual pre-training, and the performance gap diminishes rapidly as the number of labeled data increases. To better exploit the limited available data, we propose a framework encompassing a context augmentation method and sequential self-distillation. Comprehensive experiments show that given only two or more labeled samples per class, direct fine-tuning outperforms many strong baselines that utilize external data sources for continual pre-training. The findings were published in the Findings of ACL-2023 [117].

Contribution 4: new techniques to compress models under few-shot scenario. To enhance the applicability of the model in resource-constrained scenarios, we investigate model compression tailored for few-shot intent detection. We propose an effective approach using generative PLMs for data augmentation, coupled with a novel vocabulary pruning technique. Comprehensive experiments demonstrate our method's efficacy. Remarkably, we achieve a compression ratio of 21 with imperceptible loss in the performance.

Thesis organization. Chapter 2 introduces background knowledge and provides an overview of existing literature. Chapter 3 studies the cross-domain transferablity of few-shot intent detection under both supervised setting and semi-supervised setting, designs IntentBERT, the backbone for few-shot intent detection. Chapter 4 gives a thorough analysis of the anisotropy property of IntentBERT, devises the framework to enhance the model via isotropization. Two regularizers are introduced to render the feature space more isotropic and thus yield superior performance. Chapter 5 revisits continual pre-training with extra data and direct fine-tuning with the few data, presents a framework to better exploit the few data with a novel context augmentation mechanism and self-distillation, discusses the non-necessity of continual pre-training. Chapter 6 investigates model compression for few-shot intent detection,

reveals the difficulty of model compression under few-shot scenarios, proposes a model compression framework utilizing generative PLMs for data augmentation and a novel vocabulary pruning technique to significantly reduce the vocabulary size. Chapter 7 concludes the thesis and discusses future research directions.

Chapter 2

Background and Related Works

2.1 Task-oriented Dialogue System

Building machines that can interact with human beings in natural languages has been a long-standing aspiration of artificial intelligence (AI). Nowadays, such computer systems are referred to as dialogue systems, or conversational agents. These systems are divided into two principal categories according to different goals: open dialogue system and task-oriented dialogue system. The former is designed to maximize the engagement of the users during chatting, by offering recommendations, entertainment and emotional support, etc. [46] The latter, in contrast, focuses on accomplishing specific tasks in one or multiple domains, encompassing restaurant reservation, banking services and technical services, etc. [126]

The implementation of a TOD system can be categorized into two methodologies: the pipeline method and the end-to-end method. The pipeline method first constructs discrete, independent functional components and then integrates them into the conversational system, while the end-to-end method designs a single model, accepting user utterances as inputs and directly give the feedback. However, the end-to-end structure makes the system a complete black box, engendering significant uncontrolla-

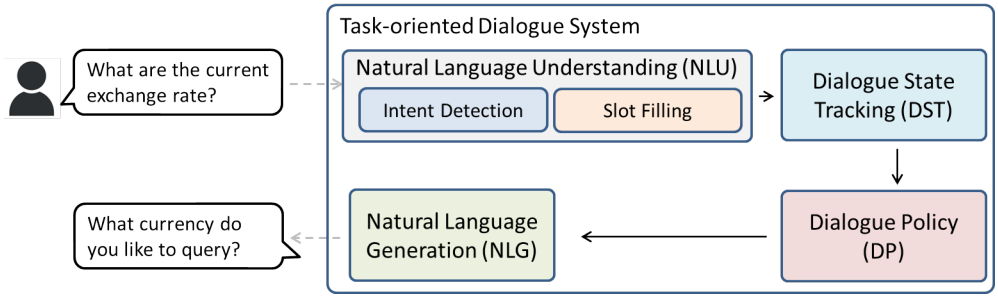


Figure 2.1: Task-oriented dialogue system.

bility of the system behavior. As a consequence, most real-world commercial systems adopt the more reliable and interpretable pipeline structure [126]. In the pipeline method, the system comprises the following components: natural language understanding (NLU), dialogue state tracking (DST), dialogue policy (DP) and natural language generation (NLG). Their functions are briefly described as follows [54, 10].

- **NLU** contains two modules, intent detection and slot filling. The two modules maps an user utterance to a structural semantic representation including intent label and slot-value label for each token.
- **DST** estimates the dialogue's goal according to the conversation history. It is typically a list of domain, slot and value, recording the users' needs.
- **DP** maps current dialogue state to an action of the system, such as database querying, order making and information confirmation.
- **NLG** generates responses in natural language according to the output of DP.

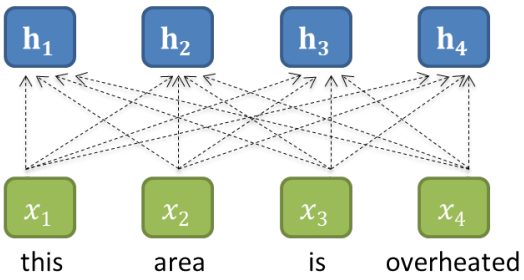
The relation of the aforementioned components is depicted in Fig. 2.1, wherein these components cooperate to realize the function of a TOD system. Intent detection is a critical functional module of NLU, aiming to detect the intents underlying users' utterances, such as currency exchange rate query. The detected intent steers the subsequent operations in TOD systems, thus exerting significant influence on the dialogue state classification, dialogue policy making, and the quality of the generated

responses [44, 116]. In this thesis, we study how to train a well-performing intent classifier with limited annotated data.

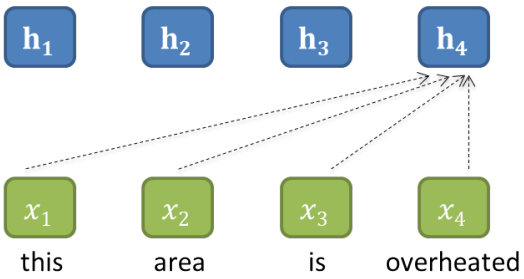
2.2 Pre-trained Language Models

PLMs' evolution. Pre-trained language models originate from the long-standing idea of distributed representation, i.e. representing the semantic meaning of a piece of text by low-dimensional vectors [24, 84, 85]. In the past decade, the development of deep learning has brought significant advancements to the semantic expressiveness of such vectors. Researchers adopt neural networks [5] instead of traditional methods such as latent semantic analysis and latent dirichlet allocation to learn these vectors, which engenders the notably superior linear regularity among words and computational efficiency [68, 128, 67]. Such advancements spawn many popular word embeddings including GloVe [75], fastText [9] and Word2Vec [67], which serve as the foundation of various NLP applications. However, these word embeddings fail to model the context-dependent nature of words [77], and thus are called *non-contextual word embeddings*. Subsequently, *contextual word embeddings* emerge. These embeddings dynamically map a word to a vector considering other words in the sentence, adopting neural networks such as Transformer [97]. These models are pre-trained with unsupervised tasks on large corpus with various tasks, including auto-regressive language modeling [113], masked language modeling [17] and sequence-to-sequence modeling [78]. Fulfilling these tasks push the neural model to learn general language knowledge that has been demonstrated to be significantly helpful for down-stream tasks. Due to the remarkable performance superiority of contextual word embeddings over non-contextual ones, currently, PLMs usually refer to the contextual ones, in accordance with the terminology used in this thesis.

Architecture. Most PLMs adopt the architecture of Transformer based on self-attention mechanism [97], including all PLMs employed in this thesis. Although other



(a) The architecture of Transformer in encoders.



(b) The architecture of Transformer in generative PLMs.

Figure 2.2: The archiecture of Transformer.

neural models have been studied, consisting of convolutional neural network [48] and recurrent models [41, 16], both of them suffer long-term dependency problem, i.e. only the local context around the word is modeled. To mitigate this issue, Transformer adopts a *fully-connected* architecture based on self-attention mechanism, as shown in Fig. 2.2a. $\mathbf{h}_i$ denotes the embedding of the i_{th} token x_i, and it is a mapping result considering all other words in the sentence. However, in generative PLMs, each token usually attends to only the preceding tokens for the generation, as shown in Fig. 2.2b.

PLMs adopted in this thesis. The study of few-shot intent detection in this thesis is based on several PLMs. The first category is encoder that encodes the semantic of utterances into vectors. We adopt two PLMs as follows.

- **BERT [17].** BERT is among the most successful PLMs. With a Transformer architecture of 12 layers and 12 attention heads, BERT is pre-trained on

BooksCorpus of 800 million words [130] and English Wikipedia of 2,500 million words, adopting masked language modeling and next sentence prediction as the pre-training tasks.

- **RoBERTa [60].** RoBERTa is a more robust version of BERT. It optimizes the hyper-parameters and pre-training configurations such as word masking scheme, pre-training objectives, batch size and corpus size. RoBERTa matches or exceed the performance of other concurrent models.

The second category is generative PLM that generates texts as follows.

- **GPT-3 [11].** GPT-3 is an advanced auto-regressive PLM with Transformer architecture, developed by OpenAI. It not only performs well on a wide spectrum of NLP tasks such as translation, question-answering without fine-tuning, but also generates text of human-level quality. OpenAI does not release the parameters of GPT-3, but provide only APIs for inference. In this thesis, we adopt the Davinci version of 175 billion parameters.

- **GPT-4 [73].** GPT-4 is the most advanced version of GPT series models with around 170 trillion parameters [51]. It generates more coherent, contextual, and appropriate text. Similar to GPT-3, GPT-4 can be accessed only via APIs.

- **GPT-J [100].** GPT-J is an open-source alternative to GPT-3 with 6 billion parameters, developed by EleutherAI. Due to smaller size, GPT-J can be deployed on consumer-grade GPUs, but with competitive zero-shot performance compared to GPT-3 of comparable size.

- **OPT-30B [123].** OPT-30B is one of the OPT series of models released by Meta AI, with 30 billion parameters, which matches the performance of the GPT-3 class of models. We adopt the above generative models to enhance the data-efficiency and also to obtain a small model for few-shot intent detection.

2.3 Few-shot Intent Detection

Traditional methods. Because of the importance of intent detection in TOD systems, it has been attracting interests from scholars. Early methods are based on rules [21] and statistical features such as n-gram features [6] and salience [33], coupled with traditional classifiers including support vector machine (SVM) [37] and boosting classifier [96]. However, rule-based methods require the expensive maintenance of the hand-crated rules. Statistical features are more convenient to obtain, but fail to effectively encode highly abstract language semantics. Meanwhile, scholars start to adopt neural models. [96] constructs deep convex networks with n-gram features. [57] and [82] adopt a recurrent neural network (RNN) encoder-decoder architecture to jointly tackle intent detection and slot filling. [111] employs a convolutional neural network (CNN) to encode local text semantics. Nevertheless, these early explorations use only a small-scale corpus to train the model, including word embeddings. Moreoever, neural models capitalize on large amount of labeled data, which is usually prohibitive to obtain [27, 98]. This limitation triggers the surge of research interests in *few-shot intent detection*, i.e. training a well-performing intent classifiers with a few annotations.

Methods based on non-contextual word embeddings. Then, a series of non-contextual word embeddings are pre-trained on large corpus and then are released for public usage, based on which researchers focus on the design of neural model architecture to tackle few-shot intent detection. Using Glove, RobustTC [115] adopts CNNs to build a clustering-based dynamic metric function, Induction network [32] introduces the capsule network and dynamic routing [86] to enhance the expressive power of intent representations. With fastText, [72] designs a sophisticated semantic matching and aggregation network to measure semantic similarity. Nonetheless, the representative power of non-contextual word embeddings is limited, and thus few-shot intent detection is still a challenge.

Methods based on contextual word embeddings. Following BERT [17], contextual word embeddings have induced a paradigm shift in NLP. To deal with few-shot intent detection, the mainstream efforts have been dedicated to adapting PLMs to intent detection by conducting continual pre-training [36, 35] on dialogue corpus or relevant tasks. [63] fine-tunes a PLM on an unlabeled dialogue corpus containing millions of conversations. [103] further pre-trains PLMs on a task-oriented dialogue corpus of $100,000$ utterances with masked language modelling (MLM). [39, 13] investigates a dual encoder model trained with response selection tasks on conversational input-response pairs. [121] conducts continual pre-training with around 1 million annotated samples for natural language inference. [122] constructs some pre-training tasks based on the wikiHow database with $110,000$ articles. [99] propose a two-stage procedure, adaptive conversational fine-tuning followed by task-tailored fine-tuning. [61] conduct continual pre-training with paraphrase dataset. Besides, [106, 108] exploit off-the-shelf BERT to generate novel utterances for intent detection. The study of few-shot intent detection has been extended to other settings, including semi-supervised learning [20, 19], incremental learning [107] and multi-label classification [43]. However, these works do not address the following issues for few-shot intent detection:

- **Transferrability**. Since TOD systems are devised for specific domains, such as the banking service and the medical domain, it is valuable if we can transfer the knowledge learned from source domains to the target one, to mitigate the scarcity of annotated data, which remains under-explored. In addition, given the easily available unlabeled utterances in the target domain, the feasibility to jointly utilize these data and the source domain data is of interests.

- **Expressiveness**. The expressiveness of PLMs is harmed by the notorious anisotropy property of the embedding space, i.e. the embeddings are distributed in a long, narrow area. It remains on open question how the fine-tuning of PLMs affects the property. The potential of mitigating the property for performance enhancement is to be unearthed.

- **Efficiency**, including *data-efficiency* and *model-efficiency*.
  - Data-efficiency. Extra data is usually required in most aforementioned methods in the intermediate stage of continual pre-training, besides the few labeled data in the target task. However, the data-efficiency may be improved if we manage to eliminate the intermediate stage without performance deterioration.
  - Model-efficiency. PLMs suffer from inferior model-efficiency due to their gigantic parameter sizes, limiting the deployment in resource-constrained computational scenarios. Model compression under the few-shot constraint is still a challenge.

In this thesis, we focus on the issues as above.

2.4 Datasets

We conduct the study with five large-scale practical benchmark datasets as follows. The dataset statistics are summarized in Table 2.1.

Dataset	#Intent	#Train	#Dev	#Test
OOS	150	15000	3000	4500
BANKING77	77	10003	0	3080
HINT3	51	1579	0	676
HWU64	64	8954	1076	1076
MCID	16	1258	148	339

Table 2.1: Dataset statistics.

OOS [52] contains labeled data of 10 domains, with 15 intents in each domain, plus out-of-scope data that do not belong to any of the intents. All data is collected via crowd-sourcing. Since this thesis focuses on intent classification, we do not use the out-of-scope data.

HWU64 [59] is a large-scale multi-domain dataset with 64 intents, collected via crowd-sourcing. Unlike OOS which is balanced across intents, different intents in HWU64 have different numbers of labeled data.

HINT3 [3] is created from real conversational systems with 51 intents over 3 domains.

BANKING77 [13] is a single-domain dataset focusing on banking service. It has 77 semantically close intents, and thus is challenging. Some intents partially overlap with others, and we can hardly rely on individual word to correctly classify the intent. This dataset requires fine-grained decision.

MCID [2] is a cross-lingual dataset for "Covid-19", generated by annotators using the ontology describing all intents with a few examples. It covers multiple languages: English, Spanish, French and German. We use only the English data.

We adopt these datasets in different ways, some of them as the transferring source while some of them for evaluation. We will go into details in each chapter.

Chapter 3

Cross-domain Transferability

3.1 Motivation

In this chapter, we investigate the cross-domain transferrability for few-shot intent detection. The problem is important because the acquisition of labeled data for a novel domain to deploy a new service is expensive, while abundant labeled data from other domains remains readily accessible. For instance, the COVID-19 pandemic engenders the development of related chatbot systems [62], and it may be possible to leverage labeled data from domains like "Banking", as shown in Fig. 3.1. On the other hand, alternative data sources for continual pre-training have been investigated by scholars, but the consumed data is often large and thus yields low data-efficiency. Below, we summarize the most relevant studies in this direction.

- **CONVBERT** [63] finetunes BERT on an unlabeled dialogue corpus consisting of nearly 700 million conversations.
- **TOD-BERT** [103] further pre-trains BERT on a task-oriented dialogue corpus of $100,000$ unlabeled samples with masked language modelling (MLM) and response contrastive objectives.

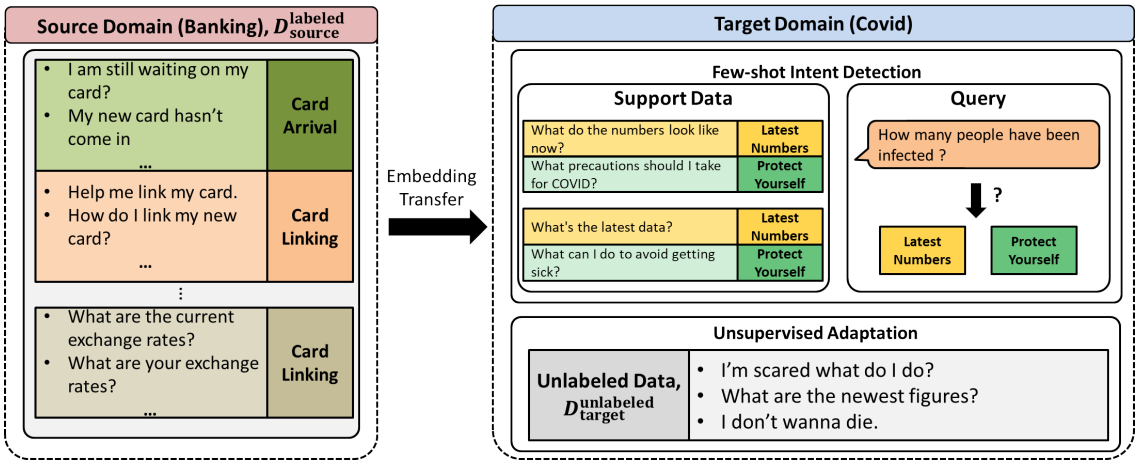


Figure 3.1: Cross-domain few-shot classification.

- **USE-ConveRT** [39, 13] investigates a dual encoder model trained with response selection tasks on 727 million input-response pairs.

- **DNNC** [121] pre-trains a language model with around 1 million annotated samples for natural language inference (NLI) and use the pre-trained model for intent detection.

- **WikiHowRoBERTa** [122] constructs some pre-training tasks based on the wikiHow database with $110,000$ articles.

To initiate the study of cross-domain few-shot intent detection, we focus on two aspects: 1) The mechanism to transfer knowledge from the source domains to the target domain, and 2) The adaptation to the target domain in the presence of a limited corpus of labeled data. For the first aspect, we propose supervised pre-training with publicly available intent detection dataset of multiple domains to learn transferablity knowledge, which generates a backbone for few-shot intent detection called IntentBERT. For the second aspect, we design a joint pre-training scheme, which simultaneously optimizes the classification error on the source labeled data and the language modeling loss on the target unlabeled data.

3.2 Method

3.2.1 Problem Definition

The objective of cross-domain few-shot intent classification is identifying novel intent classes within the target domain, leveraging only a few labeled samples. We assume the existence of labeled data in the source domains of a different label space, $\mathcal{D}_{\text{source}}^{\text{labeled}} = \{(x_i, y_i)\}$, where y_i is the label of utterance x_i. The source and target domains may have *very different semantics*. To illustrate, we give an example in Figure 3.1, wherein the source domain is "Banking", whereas the target domain is "Covid". Moreover, to adapt to the target domain, we propose exploiting the easily available unlabeled data in the target domain, $\mathcal{D}_{\text{target}}^{\text{unlabeled}} = \{x_i\}$, as shown in Fig. 3.1.

3.2.2 Supervised Pre-training and Joint Pre-training

Our pre-training method relies on the existence of $\mathcal{D}_{\text{source}}^{\text{labeled}}$. Such data samples can be readily obtained from public intent detection datasets such as OOS [52] and HWU64 [59]. As will be shown in the experiments, roughly $1,200$ examples from either OOS or HWU64 are enough for the pre-trained intent detection model to achieve a superior performance on drastically different target domains such as "Covid". Given additional $\mathcal{D}_{\text{target}}^{\text{unlabeled}}$, we propose a joint pre-training scheme that is empirically proven to be highly effective.

Supervised pre-training. Given $\mathcal{D}_{\text{source}}^{\text{labeled}} = \{(x_i, y_i)\}$ with N different classes, we employ a simple method to fine-tune BERT. Specifically, a linear layer is attached on top of BERT as the classifier, i.e.,

$$p(y|h_i) = \text{softmax}\left(\mathbf{W}h_i + \mathbf{b}\right) \in \mathbb{R}^N, \tag{3.1}$$

where $h_i \in \mathbb{R}^d$ is the feature representation of x_i. We use the feature vector of the [CLS] token to represent the sentence. $\mathbf{W} \in \mathbb{R}^{N \times d}$ and $\mathbf{b} \in \mathbb{R}^N$ are parameters of

the linear layer. Model parameters $\theta = \{\phi, \mathbf{W}, \mathbf{b}\}$, with ϕ being the parameters of BERT, are trained on $\mathcal{D}_{\text{source}}^{\text{labeled}}$ with a cross-entropy loss:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{ce}\left(\mathcal{D}_{\text{source}}^{\text{labeled}}; \theta\right). \tag{3.2}$$

After training, the fine-tuned BERT is expected to have learned general intent detection skills, and hence we call it IntentBERT.

Joint pre-training. Given unlabeled target data $\mathcal{D}_{\text{target}}^{\text{unlabeled}}$, we can leverage it to further enhance our IntentBERT, by simultaneously optimizing a language modeling loss on $\mathcal{D}_{\text{target}}^{\text{unlabeled}}$ and the supervised loss in Eq. (3.2). The language modeling loss can help to learn semantic representations of the target domain while preventing overfitting to the source data. Specifically, we use MLM as the language modeling loss, in which a proportion of input tokens are masked with the special token [MASK] and the model is trained to retrieve the masked tokens. The joint training loss is formulated as:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{ce}(\mathcal{D}_{\text{source}}^{\text{labeled}}; \theta) + \lambda \mathcal{L}_{mlm}(\mathcal{D}_{\text{target}}^{\text{unlabeled}}; \theta), \tag{3.3}$$

where λ is a hyperparameter. It is used to balance the supervised loss and the unsupervised loss.

Few-shot intent classification. After pre-training, the parameters of IntentBERT are fixed, and it can be immediately used as a feature extractor for novel few-shot intent classification tasks. The classifier can be a parametric one such as logistic regression or a non-parametric one such as nearest neighbor. A parametric classifier will be trained with the few labeled examples provided in a task and make predictions on the unlabeled queries. As will be shown in the experiments, a simple linear classifier suffices to achieve very good performance, thanks to the effective utterance representations produced by IntentBERT.

3.3 Experiment

3.3.1 Setup

Datasets. To train IntentBERT, we continue to pre-train BERT on either **OOS**[1] or **HWU64**. Both datasets contain multiple domains, providing rich resources to learn the general intent detection knowledge[2]. For evaluation, we employ three datasets: **BANKING77**, **MCID** and **HINT3**.

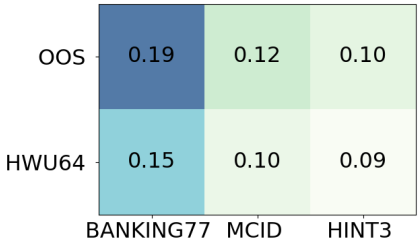


Figure 3.2: Vocabulary overlap.

Fig. 3.2 visualizes the vocabulary overlap between the source training data and target test data, which is calculated as the proportion of the shared words in the combined vocabulary of any two datasets after removing stop words. It is observed that the overlaps are quite small, indicating the existence of large semantic gaps.

Evaluation. The classification performance is evaluated by C-way K-shot tasks. For each task, We randomly sample C classes and K examples per class to train the classifier, and then we sample extra 5 examples per class as queries for evaluation. The accuracy is averaged over 500 such tasks.

Baselines. We compare IntentBERT to the following strong baselines. **BERT-**

[1]The domains "Banking" and "Credit Cards" are excluded because they are semantically close to the evaluation data.

[2]We have also experimented with the combination of both datasets but observed no better results.

Method	$\mathcal{D}_{\text{target}}^{\text{unlabeled}}$	BANKING77	MCID	HINT3
BERT-Freeze	✗	52.62(12.41)	57.84(11.72)	47.3(12.06)
CONVBERT	✗	68.27(12.34)	67.7(11.54)	72.61(10.90)
TOD-BERT	✗	77.66(7.35)	64.10(9.01)	68.9(11.69)
DNNC	✗	67.54(15.40)	56.22(16.70)	64.08(14.77)
WikiHowRoBERTa	✗	34.92(10.52)	30.82(9.93)	31.72(10.34)
IntentBERT (HWU64) (ours)	✗	**78.38(10.55)**	**74.54(11.89)**	**77.91(10.64)**
IntentBERT (OOS) (ours)	✗	**82.44(8.31)**	**77.12(9.02)**	**80.09(10.40)**
IntentBERT (OOS)+MLM (ours)	✓	**88.91(8.98)**	**86.30(9.84)**	**87.12(9.75)**

(a) Main results for 5-way 2-shot tasks.

Method	$\mathcal{D}_{\text{target}}^{\text{unlabeled}}$	BANKING77	MCID	HINT3
BERT-Freeze	✗	69.95(11.7)	72.43(10.72)	66.80(10.45)
CONVBERT	✗	86.55(8.18)	83.52(7.93)	87.20(7.88)
TOD-BERT	✗	89.44 (5.13)	77.72(11.08)	83.52(8.55)
USE-ConveRT¶	✗	85.22	–	–
DNNC	✗	89.84(7.53)	80.01(9.92)	87.85(8.08)
WikiHowRoBERTa	✗	41.60(10.10)	36.36(9.68)	39.02(9.88)
IntentBERT (HWU64) (ours)	✗	**90.02(7.47)**	**85.92(8.82)**	**89.42(7.94)**
IntentBERT (OOS) (ours)	✗	**91.84(4.22)**	**88.12(5.90)**	**90.18(7.38)**
IntentBERT (OOS)+MLM (ours)	✓	**95.22(5.14)**	**92.40(6.16)**	**94.02(5.98)**

(b) Main results for 5-way 10-shot tasks.

Table 3.1: Main results for 5-way tasks. ¶ stands for results from the original paper.

Freeze simply freeze the off-the-shelf BERT; **TOD-BERT** [103] further pre-trains BERT on a huge amount of task-oriented conversations with MLM and response selection tasks; **CONVBERT** [63] further pre-trains BERT on a large open-domain multi-turn dialogue corpus; **USE-ConveRT** [39, 13] is a fast embedding-based classifier pre-trained on an open-domain dialogue corpus by dialogue response selection tasks; **DNNC** [121] further pre-trains a BERT-based model on NLI tasks and then applies a similarity-based classifier for classification; **WikiHowRoBERTa** [122] fur-

ther pre-trains RoBERTa [60] on fake intent detection data synthesized from wikiHow[3]. All the baselines (except BERT-Freeze) adopt a second pre-training stage, but with different objectives and on different corpus. In our experiments, all the baselines (except DNNC) use logistic regression as the classifier. For DNNC, we strictly follow the original implementation[4] to pre-train a BERT-style pairwise encoder to estimate the best matched training example for a query utterance.

Training details. We use $\text{BERT}_{\text{base}}$[5] (the base configuration with $d = 768$) as the encoder, Adam [49] as the optimizer, and PyTorch library for implementation. The model is trained with Nvidia GeForce RTX 2080 Ti GPUs. For supervised pre-training, we use validation to control early-stop to prevent overfitting. Specifically, we use HWU64 for validation when pre-training with OOS and vice versa. The training is stopped if no improvement in accuracy is observed in 3 epochs. For joint pre-training, λ is set to 1. The number of training epochs is fixed to 10, since it is not prone to overfitting.

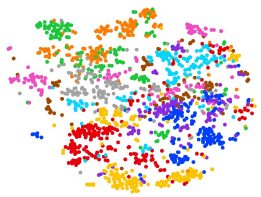
3.3.2 Results

Main results. The main results are provided in Table 3.1. First, IntentBERT (either pre-trained with OOS or HWU64) consistently outperforms all the baselines by a significant margin in most cases. Take the results of 5-way 2-shot classification on MCID for example, IntentBERT (OOS) outperforms the strongest baseline CONVBERT by an absolute margin of 9.4%, demonstrating the high effectiveness of our pre-training method. The cross-domain transferability of IntentBERT indicates that despite semantic domain gaps, most intent detection tasks probably share a similar underlying structure, which could be learned with a small set of labeled utterances. Second, IntentBERT (OOS) seems to be more effective than IntentBERT (HWU64),

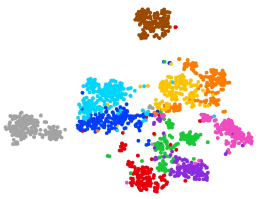
[3]https://www.wikihow.com/

[4]https://github.com/salesforce/DNNC-few-shot-intent

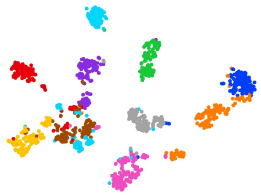
[5]https://github.com/huggingface/transformers



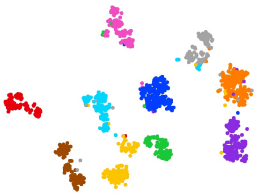
(a) BERT



(b) TOD-BERT



(c) IntentBERT (ours)



(d) IntentBERT+MLM (ours)

Figure 3.3: Visualization of the embedding spaces with t-SNE.

which may be due to the semantic diversity of the training corpus. Nevertheless, the small difference in performance between them shows that our pre-training method is not sensitive to the training corpus.

Finally, our proposed joint pre-training scheme (Section 3.2.2) achieves significant improvement over IntentBERT (up to 9.2% absolute margin), showing the high effectiveness of joint pre-training when target unlabeled data is accessible.

3.3.3 Analysis

Visualization To obtain deeper understanding of the quality of the feature space generated by the proposed methods, we visualize the space of 10 randomly sampled classes with 500 data per class from BANKING77 in Fig. 3.3, comparing our methods to strong baselines. The figure clearly demonstrates the superiority of our pre-trained models, echoing the quantitative evaluation results in Section 3.3.2.

Amount of labeled data for pre-training. We reduce the data used for pre-training in two dimensions: the number of domains and the number of samples per

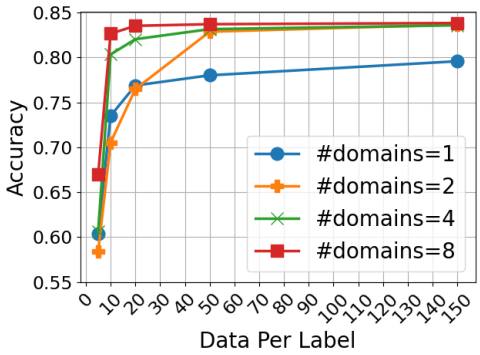


Figure 3.4: The impact of the labeled data quantity for pre-training.

class. We randomly sample 1, 2, 4 and 8 domains for multiple times and report the averaged results in Fig. 3.4. The source domain is OOS dataset and the results are evaluated on 5-way 2-shot tasks on BANKING77. It is found that the training data can be dramatically reduced without harming the performance. The model trained on 8 domains and 10 samples per class performs on par with that on 8 domains and 150 samples per class. In general, we need only around $1,200$ annotated utterances to train IntentBERT, which can be easily obtained in public datasets. This finding indicates that using small task-relevant data for pre-training may be a more effective and efficient fine-tuning paradigm.

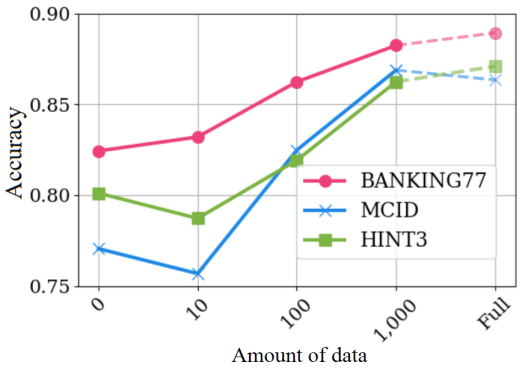


Figure 3.5: The impact of the unlabeled data quantity.

Amount of unlabeled data for joint pre-training. We randomly sample a fraction of unlabeled utterances in the target domain and re-run the joint training. The

results are evaluated on 5-way 2-shot tasks with OOS as the source dataset. As shown in Fig. 3.5, the accuracy keeps increasing when the number of unlabeled samples grows from 10 to 1,000 and tends to saturate after reaching 1,000. Surprisingly, 1,000 utterances in BANKING77 can yield a comparable performance than the full dataset (13,083 utterances). Generally, it does not need much unlabeled data to reach a high accuracy.

Methods	BANKING77	MCID	HINT3
BERT→MLM(target)	80.52	62.96	72.43
IntentBERT→MLM(target)	82.04	75.91	77.92
IntentBERT+MLM(source)	84.08	75.88	78.49
IntentBERT+MLM(target)	**88.92**	**86.34**	**87.11**

Table 3.2: Ablation study on joint pre-training. → denotes moving to the next training stage. + denotes joint optimization of both loss functions.

Ablation study on joint pre-training. First, we investigate a two-stage pre-training scheme [36] where we use BERT or IntentBERT as initialization and perform MLM in the target domain (the top two rows in Table 3.2). In the table, the data used for the experiment (either from "target" or "source") is shown in the brackets. It can be seen that they perform much worse than our joint pre-training scheme (the bottom row). Second, we use the source data instead of the target data for MLM in joint pre-training (the third row), and observe consistent performance drops, which shows the necessity of a task-specific corpus. The experiment is conducted with 5-way 2-shot tasks using OOS as the source dataset. The result echoes the findings in [36] which demonstrates the effectiveness of continual pre-training over domain-specific corpora or the unlabeled data of target tasks. However, our result of IntentBERT+MLM(source) shows that even across domains and label set, the unlabeled utterances with underlying intents are still significantly beneficial.

3.4 Conclusion

In this chapter, we give a comprehensive empirical study into cross-domain few-shot intent detection. We have proposed knowledge transferring methods and adaptation methods. Extensive experiments have shown the superior performance compared to competitive baselines on various benchmark datasets. IntentBERT is developed. It is a backbone network for few-shot intent detection, which is obtained by fine-tuning BERT using publicly available labeled utterances. The results demonstrate the transferability across domains for few-shot intent detection. The proposed method is adopted to facilitate intent discovery task [125].

Chapter 4

Mitigating Anisotropy for Expressiveness

4.1 Motivation

In Chapter 3, we devise IntentBERT, a backbone for few-shot intent detection which utilizes public intent datasets for continual pre-training of BERT. However, as will be shown in this chapter, IntentBERT suffers from severe anisotropy, an undesirable property of PLMs [30, 25, 55].

Anisotropy is a geometric property that semantic vectors fall into a long, narrow cone. It has been identified as a crucial factor for the sub-optimal performance of PLMs on a variety of downstream tasks [30, 4, 12, 25], which is also known as the representation degeneration problem [30]. Fortunately, isotropization techniques can adjust the embedding space, and thus yields the significant performance improvement [91, 79].

Hence, this chapter aims to answer the question:

- Is it possible to improve supervised pre-training via *isotropization* for few-shot intent detection?

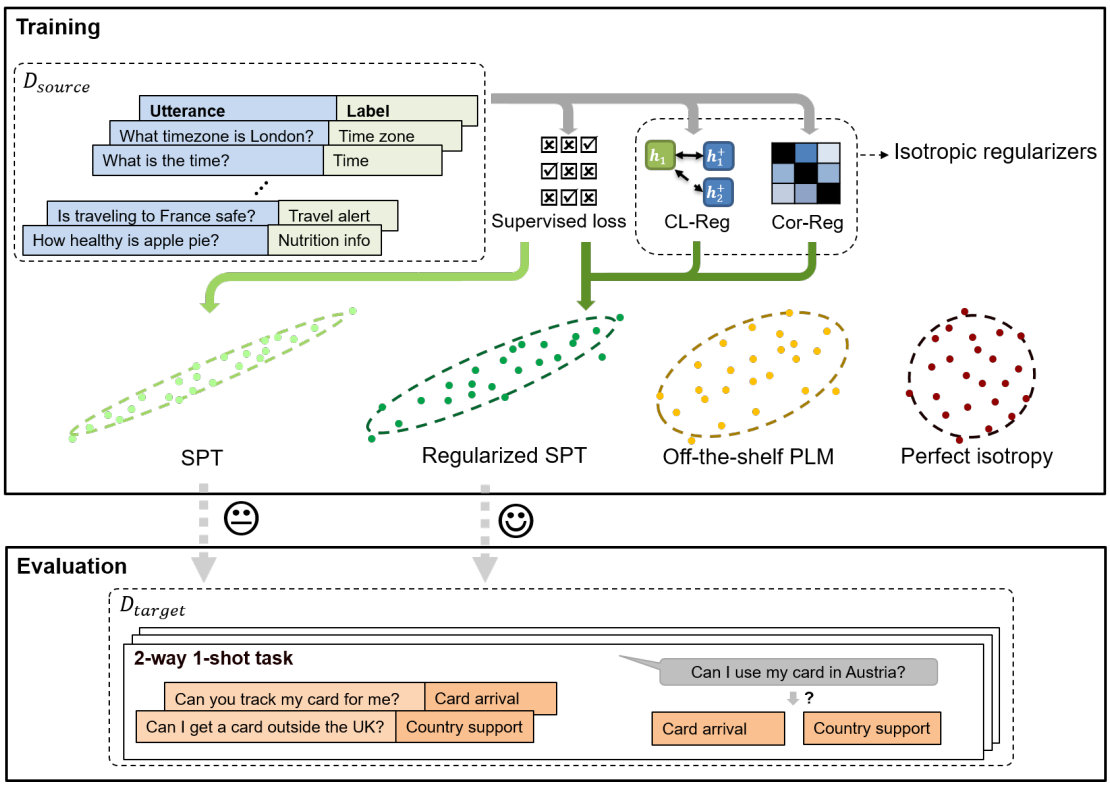


Figure 4.1: Illustration of our proposed regularized supervised pre-training.

Many isotropization techniques have been developed based on transformation [91, 45], contrastive learning [31], and top principal components elimination [71]. However, these methods are designed for off-the-shelf PLMs. When applied on PLMs that have been fine-tuned on down-stream NLP tasks such as semantic textual similarity task or intent classification, they may introduce an adverse effect, as observed in [81] and our pilot experiments.

In this chapter, we present a comprehensive study on the isotropy property of PLMs for few-shot intent detection. Specifically, we first study the interaction of supervised pre-training and isotropization, and then propose to regularize supervised pre-training with isotropic regularizer. The idea is illustrated in Fig. 4.1, wherein SPT denotes supervised pre-training (fine-tuning an off-the-shelf PLM on a set of labeled utterances), which makes the feature space more anisotropic. we devise two regularizers, a

contrastive-learning-based regularizer (CL-Reg) and a correlation-matrix-based regularizer (Cor-Reg), each of which can increase the isotropy of the feature space during supervised training. Extensive evaluation and analysis are conducted to validate the effectiveness of the proposed approach.

4.2 Interaction of Supervised Pre-training and Isotropization

To deepen our understanding of the isotropy property, we first conduct pilot experiments to gain some insights into the interaction between isotropization and the supervised continual pre-training which generates IntentBERT.

4.2.1 Measuring isotropy

Following [71, 8], we adopt the following measurement of isotropy:

$$\mathrm{I}(\mathbf{V}) = \frac{\min_{\mathbf{c} \in C} \mathrm{Z}(\mathbf{c}, \mathbf{V})}{\max_{\mathbf{c} \in C} \mathrm{Z}(\mathbf{c}, \mathbf{V})}, \tag{4.1}$$

where $\mathbf{V} \in \mathbb{R}^{N \times d}$ is the matrix of stacked embeddings of N utterances (note that the embeddings have zero mean), C is the set of unit eigenvectors of $\mathbf{V}^\top \mathbf{V}$, and $\mathrm{Z}(\mathbf{c}, \mathbf{V})$ is the partition function [4] defined as:

$$\mathrm{Z}(\mathbf{c}, \mathbf{V}) = \sum_{i=1}^{N} \exp\left(\mathbf{c}^\top \mathbf{v}_i\right), \tag{4.2}$$

where $\mathbf{v}_i$ is the i_{th} row of $\mathbf{V}$. $\mathrm{I}(\mathbf{V}) \in [0, 1]$, and 1 indicates perfect isotropy.

4.2.2 Fine-tuning Leads to Anisotropy

To observe the impact of fine-tuning on isotropy, we follow Chapter 3 to fine-tune BERT [17] with standard supervised training on a small set of an intent detection

Dataset	BERT	IntentBERT
BANKING	0.96	$0.71_{(0.04)}$
HINT3	0.95	$0.72_{(0.03)}$
HWU64	0.96	$0.72_{(0.04)}$

Table 4.1: The impact of fine-tuning on isotropy.

benchmark OOS [52] (details are given in Chapter 3). We then compare the isotropy of the original embedding space (BERT) and the embedding space after fine-tuning (IntentBERT) on target datasets. As shown in Table 4.1, after fine-tuning, the isotropy of the embedding space is notably decreased on all datasets. Hence, it can be seen that *fine-tuning may render the feature space more anisotropic*. In the table, the mean and standard deviation of 5 runs are reported.

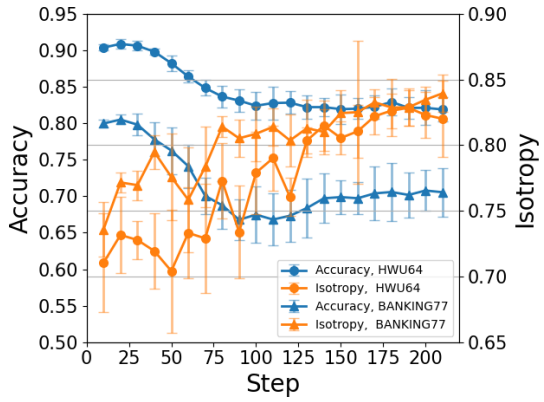
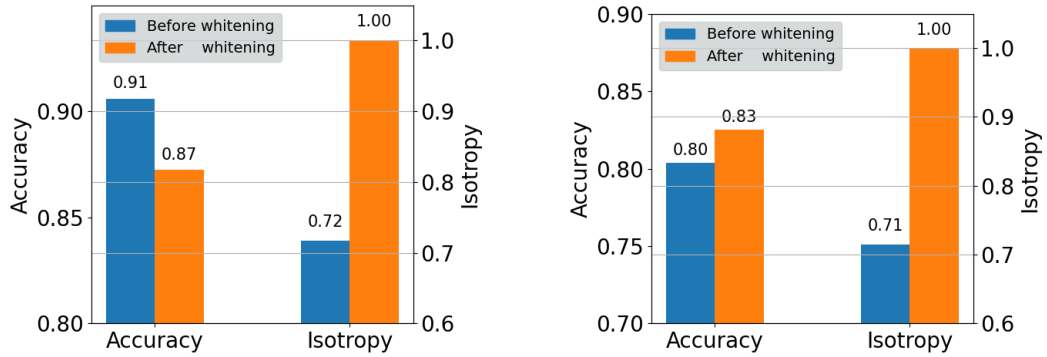


Figure 4.2: The impact of contrastive learning on IntentBERT with experiments on HWU64 and BANKING77 datasets.

4.2.3 Isotropization after Fine-tuning May Be Harmful

To examine the effect of isotropization on a fine-tuned model, we apply two strong isotropization techniques to IntentBERT: dropout-based contrastive learning [31] and whitening transformation [91]. The former fine-tunes PLMs in a contrastive learn-



(a) HWU64. (b) BANKING77.

Figure 4.3: The impact of whitening on IntentBERT with experiments on HWU64 and BANKING77 datasets.

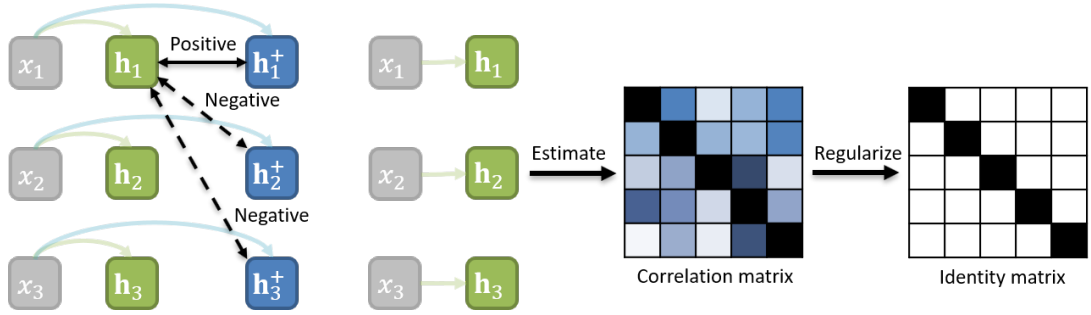
ing manner[1], while the latter transforms the semantic feature space into an isotropic space via matrix transformation. These methods have been demonstrated highly effective [31, 91] when applied to off-the-shelf PLMs, but things are different when they are applied to fine-tuned models. As shown in Fig. 4.2, contrastive learning improves isotropy (orange lines), but it significantly lowers the performance ((blue lines) on two benchmarks. As for whitening transformation, it has inconsistent effects on the two datasets, as shown in Fig. 4.3. It hurts the performance on HWU64 (Fig. 4.3a) but yields better results on BANKING77 (Fig. 4.3b), while producing nearly perfect isotropy on both. The above observations indicate that *isotropization may hurt fine-tuned models*, which echoes the recent finding in [80].

4.3 Method

The study in Section 4.2 reveals the anisotropy of a PLM fine-tuned on intent detection tasks and the challenge of applying isotropization techiniques to the fine-tuned model. To mitigate the anisotropy issue, in this section, we propose a joint fine-tuning

[1]We refer the reader to [31] for details.

and isotropization framework. Specifically, we propose two regularizers to make the feature space more isotropic during fine-tuning.



(a) CL-Reg. (b) Cor-Reg.

Figure 4.4: Illustration of CL-Reg (contrastive-learning-based regularizer) and Cor-Reg (correlation-matrix-based regularizer).

4.3.1 Problem Definition

Few-shot intent detection targets to train a good intent classifier with only a few labeled data $\mathcal{D}_{\text{target}} = \{(x_i, y_i)\}_{N_t}$, where N_t is the number of labeled samples in the target dataset, x_i denotes the i_{th} utterance, and y_i is the label. To tackle the problem, we have proposed to learn intent detection skills by the continual supervised pre-training on a small subset of public intent detection benchmarks in Chapter 3. $\mathcal{D}_{\text{source}} = \{(x_i, y_i)\}_{N_s}$ denotes the source data used for pre-training, where N_s is the number of examples. After the training is finished, the PLM can be directly used on the target dataset by attaching a readily available classifier such as a linear regression classifier, or a support vector machine. It has been shown in Chapter 3 that this method can work well when even the label spaces of $\mathcal{D}_{\text{source}}$ and $\mathcal{D}_{\text{target}}$ are disjoint. However, our analysis in Section 4.2 uncover a critical limitation of supervised pre-training, i.e. it makes the anisotropy property of the feature space worse.

4.3.2 Regularizing Supervised Pre-training with Isotropization

To mitigate the anisotropy of the PLM fine-tuned by supervised pre-training, we propose a joint training objective by adding a regularization term $\mathcal{L}_{\text{reg}}$ for isotropization:

$$\mathcal{L} = \mathcal{L}_{\text{ce}}(\mathcal{D}_{\text{source}}; \theta) + \lambda \mathcal{L}_{\text{reg}}(\mathcal{D}_{\text{source}}; \theta), \tag{4.3}$$

where λ is a weight parameter. The aim is to learn intent detection skills while maintaining an appropriate degree of isotropy. We devise two different regularizers introduced as follows.

Contrastive-learning-based Regularizer. Inspired by the recent success of contrastive learning in mitigating anisotropy [112, 31], we employ the dropout-based contrastive learning loss used in [31] as the regularizer:

$$\mathcal{L}_{\text{reg}} = -\frac{1}{N_b} \sum_{i}^{N_b} \log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N_b} e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}. \tag{4.4}$$

In particular, $\mathbf{h}_i \in \mathbb{R}^d$ and $\mathbf{h}_i^+ \in \mathbb{R}^d$ are two different representations of utterance x_i generated by the PLM with built-in standard dropout [90], i.e., x_i is passed to the PLM twice with different dropout masks to produce $\mathbf{h}_i$ and $\mathbf{h}_i^+$. $\text{sim}(\mathbf{h}_1, \mathbf{h}_2)$ denotes the cosine similarity between $\mathbf{h}_1$ and $\mathbf{h}_2$. τ is the temperature parameter. N_b is the batch size. Since $\mathbf{h}_i$ and $\mathbf{h}_i^+$ represent the same utterance, they form a positive pair. Similarly, $\mathbf{h}_i$ and $\mathbf{h}_j^+$ form a negative pair, since they represent different utterances. An example is given in Fig. 4.4a. x_i is the i_{th} utterance in a batch of size 3, and is fed to the PLM twice with built-in dropout to produce two different representations of x_i: $\mathbf{h}_i$ and $\mathbf{h}_i^+$. Positive and negative pairs are then constructed for each x_i, i.e. $\mathbf{h}_1$ and $\mathbf{h}_1^+$ form a positive pair for x_1, while $\mathbf{h}_1$ and $\mathbf{h}_2^+$, $\mathbf{h}_1$ and $\mathbf{h}_3^+$, form negative pairs for x_1. By minimizing the contrastive loss, positive pairs are pulled together while negative pairs are pushed away, which in theory enforces an isotropic feature space [31]. In [31], the contrastive loss is used as the single objective to fine-tune

off-the-shelf PLMs in an unsupervised manner, while in this work we use it jointly with supervised pre-training to fine-tune PLMs for few-shot learning.

Correlation-matrix-based Regularizer. The above regularizer enforces isotropization implicitly. Here, we propose a new regularizer that explicitly enforces isotropization. The perfect isotropy is characterized by zero covariance and uniform variance [91, 129], i.e., a covariance matrix with uniform diagonal elements and zero non-diagonal elements. Isotropization can be achieved by endowing the feature space with such statistical property. However, as will be shown later, it is difficult to determine the appropriate scale of variance. Therefore, we base the regularizer on *correlation matrix* :

$$\mathcal{L}_{\text{reg}} = \|\mathbf{\Sigma} - \mathbf{I}\|, \tag{4.5}$$

where $\|\cdot\|$ denotes Frobenius norm, $\mathbf{I} \in \mathbb{R}^{d\times d}$ is identity matrix, $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ is the correlation matrix. $\mathbf{\Sigma}_{ij}$ denotes Pearson correlation coefficient between the i_{th} dimension and the j_{th} dimension. As shown in Fig. 4.4b, $\mathbf{\Sigma}$ is estimated with $\mathbf{h}_i$, the representations of utterances in the current batch. By pushing the correlation matrix towards identity matrix during training, we can learn a more isotropic feature space.

Moreover, the proposed two regularizers can be used together as follows:

$$\mathcal{L} = \mathcal{L}_{\text{ce}}(\mathcal{D}_{\text{source}}; \theta) + \lambda_1 \mathcal{L}_{\text{cl}}(\mathcal{D}_{\text{source}}; \theta) + \lambda_2 \mathcal{L}_{\text{cor}}(\mathcal{D}_{\text{source}}; \theta), \tag{4.6}$$

where λ_1 and λ_2 are the weight parameters, and $\mathcal{L}_{\text{cl}}$ and $\mathcal{L}_{\text{cor}}$ denote CL-Reg and Cor-Reg, respectively. Our experiments show that better performance is often observed when they are used together.

4.4 Experiments

4.4.1 Setup

Datasets. To perform supervised pre-training, we follow Section 3.3 to adopt the **OOS** dataset. This dataset contains diverse semantics of 10 domains, and thus provide extensive knowledge of intent representation to learn. Also following the configuration in Section 3.3, we exclude the domains "Banking" and "Credit Cards" since they are similar in semantics to one of the test dataset BANKING77. We then use 6 domains for training and 2 for validation, as shown in Table 4.2. For evaluation, we employ four datasets: **BANKING77**, **HINT3**, **MCID** and **HWU64**.

Training	Validation
"Auto commute", "Work", "Home", "Meta", "Small talk", "Utility"	"Travel", "Kitchen dining"

Table 4.2: Split of domains in OOS.

Our Method. Our method can be applied to fine-tune any PLM. We conduct experiments on two popular PLMs, BERT [17] and RoBERTa [60]. For both of them, the embedding of [CLS] is used as the utterance representation. We employ logistic regression as the classifier. We select the hyperparameters $\lambda, \lambda_1, \lambda_2$, and τ by validation. The best hyperparameters are provided in Table 4.3.

Baselines. We compare our method to the following baselines. First, for BERT-based methods, **CONVBERT** [63], **TOD-BERT** [103], and **DNNC-BERT** [121] further pre-train BERT on conversational corpus or natural language inference tasks. **USE-ConveRT** [39, 13] is a transformer-based dual-encoder pre-trained on conversational corpus. **CPFT-BERT** is the re-implemented version of CPFT [120], by further pre-training BERT in an unsupervised manner with mask-based contrastive learning and masked language modeling on the same training data as ours.

Method	Hyperparameter
CL-Reg	$\lambda = 1.7, \tau = 0.05$
Cor-Reg	$\lambda = 0.04$
CL-Reg + Cor-Reg	$\lambda_1 = 1.7, \lambda_2 = 0.04, \tau = 0.05$

(a) BERT-based.

Method	Hyperparameter
CL-Reg	$\lambda = 2.9, \tau = 0.05$
Cor-Reg	$\lambda = 0.06$
CL-Reg + Cor-Reg	$\lambda_1 = 2.9, \lambda_2 = 0.13, \tau = 0.05$

(b) RoBERTa-based.

Table 4.3: Hyperparameters selected via validation.

IntentBERT-ReImp is the re-implemented version of IntentBERT as in Chapter3, which uses the same random seed, training data, and validation data as our methods for a fair comparison. For RoBERTa-based baselines, **WikiHowRoBERTa** [122] further pre-trains RoBERTa on synthesized intent detection data. **DNNC-RoBERTa** and **CPFT-RoBERTa** are similar to DNNC-BERT and CPFT-BERT except the PLM. **IntentRoBERTa** is the re-implemented version of IntentBERT based on RoBERTa, with uses the same random seed, training data, and validation data as our method. Finally, to show the superiority of the joint fine-tuning and isotropization, we compare our method against whitening transformation [91]. **BERT-White** and **RoBERTa-White** apply the transformation to BERT and RoBERTa, respectively. **IntentBERT-White** and **IntentRoBERTa-White** apply the transformation to IntentBERT-ReImp and IntentRoBERTa, respectively.

All baselines use logistic regression as classifier except DNNC-BERT and DNNC-RoBERTa, wherein we follow the original work[2] to train a pairwise encoder for nearest neighbor classification.

[2]https://github.com/salesforce/DNNC-few-shot-intent

Method	BANKING77	HINT3	HWU64	MCID	Val.
CONVBERT	68.27	72.61	81.75	67.70	90.54
TOD-BERT	77.66	68.90	83.24	64.10	88.10
DNNC-BERT	67.54	64.08	73.97	56.22	72.98
CPFT-BERT	72.09	74.34	83.02	72.16	89.33
IntentBERT-ReImp	80.38(.35)	77.09(.89)	90.61(.44)	76.67(.18)	93.62(.38)
BERT-White	72.95	65.70	75.98	65.12	87.33
IntentBERT-White	82.52(.26)	78.50(.59)	87.24(.18)	75.05(.57)	**94.89(.21)**
CL-Reg (ours)	**83.45(.35)**	**79.30(.87)**	**91.46(.15)**	**78.13(.91)**	94.43(.22)
Cor-Reg (ours)	**83.94(.45)**	**80.16(.71)**	**90.75(.35)**	**77.65(1.2)**	**95.02(.22)**
CL-Reg + Cor-Reg (ours)	**85.21(.58)**	**81.20(.45)**	**90.66(.42)**	**78.20(.78)**	**95.41(.25)**

(a) 2-shot results.

Method	BANKING77	HINT3	HWU64	MCID	Val.
CONVBERT	86.60	87.20	92.55	83.52	96.82
TOD-BERT	89.40	83.50	91.56	77.72	96.39
USE-ConveRT	85.20	–	85.90	–	–
DNNC-BERT	89.80	87.90	90.71	80.01	95.23
CPFT-BERT	89.82	90.37	93.66	81.95	97.30
IntentBERT-ReImp	92.35(.12)	89.55(.63)	95.21(.15)	87.02(.66)	97.80(.18)
BERT-White	88.86	85.70	91.26	82.07	96.05
IntentBERT-White	92.29(.33)	90.14(.26)	94.42(.08)	86.52(.06)	98.07(.12)
CL-Reg (ours)	**93.66(.22)**	**91.06(.30)**	**95.84(.12)**	**88.44(.51)**	**98.43.02)**
Cor-Reg (ours)	**93.98(.26)**	**91.38(.55)**	**95.82(.14)**	**88.53(.78)**	**98.47(.07)**
CL-Reg + Cor-Reg (ours)	**94.68(.01)**	**92.38(.01)**	**95.84(.19)**	**89.19(.29)**	**98.58(.01)**

(b) 10-shot results.

Table 4.4: 5-way evaluation results using BERT. The top 3 results are highlighted.

Training Details. We use PyTorch library and Python to build our model. We employ Hugging Face implementation[3] of *bert-base-uncased* and *roberta-base*. We use Adam [49] as the optimizer with learning rate of $2e-05$ and weight decay of $1e-03$.

[3]https://github.com/huggingface/transformers

The model is trained with Nvidia RTX 3090 GPUs. The training is early stopped if no improvement in validation accuracy is observed for 100 steps. The same set of random seeds, $\{1, 2, 3, 4, 5\}$, is used for IntentBERT-ReImp, IntentRoBERTa, and our method.

Method	BANKING77	HINT3	HWU64	MCID	Val.
WikiHowRoBERTa	32.88	31.92	30.81	26.95	34.10
DNNC-RoBERTa	74.32	68.06	69.87	62.10	58.51
CPFT-RoBERTa	80.27(.11)	79.98(.11)	83.18(.11)	70.75	86.71(.10)
IntentRoBERTa	81.38(.66)	78.20(1.72)	**90.48(.69)**	76.23(.89)	95.33(.54)
RoBERTa-White	79.27	73.13	82.65	67.51	89.90
IntentRoBERTa-White	83.75(.45)	79.64(1.38)	86.52(1.33)	74.90(1.15)	96.06(.58)
CL-Reg	**84.63(.68)**	**81.10(.49)**	**91.67(.20)**	**78.63(.95)**	**96.32(.14)**
Cor-Reg	**86.92(.71)**	**82.20(.48)**	**91.10(.18)**	**76.65(.70)**	**96.82(.03)**
CL-Reg + Cor-Reg	**87.96(.31)**	**83.55(.30)**	90.47(.39)	**77.95(.84)**	**96.35(.19)**

(a) 2-shot results.

Method	BANKING77	HINT3	HWU64	MCID	Val.
WikiHowRoBERTa	59.50	54.18	52.47	48.55	60.59
DNNC-RoBERTa	87.30	82.34	80.22	78.33	74.46
CPFT-RoBERTa	93.91(.06)	**92.55(.07)**	92.82(.06)	82.45(.12)	96.45(.05)
IntentRoBERTa	92.68(.24)	89.01(1.07)	94.49(.43)	87.27(.50)	98.32(.15)
RoBERTa-White	93.00	89.02	94.00	84.62	97.14
IntentRoBERTa-White	92.68(.31)	90.13(.66)	93.82(.53)	86.59(.68)	98.35(.21)
CL-Reg	**94.43(.34)**	91.65(.13)	**95.44(.28)**	**89.27(.38)**	**98.79(.05)**
Cor-Reg	**95.07(.41)**	**92.11(.41)**	**95.69(.12)**	**89.12(.29)**	**98.89(.03)**
CL-Reg + Cor-Reg	**95.85(.02)**	**93.17(.23)**	**95.64(.28)**	**89.80(.28)**	**98.85(.07)**

(b) 10-shot results.

Table 4.5: 5-way evaluation results using RoBERTa. The top 3 results are highlighted.

Evaluation. The baselines and our method are evaluated on C-way K-shot tasks. For each task, we randomly sample C classes and K examples per class. The $C \times K$

labeled examples are used to train the logistic regression classifier. Note that we do not further fine-tune the PLM using the labeled data of the task. We then sample another 5 examples per class as queries. Fig. 4.1 gives an example with $C = 2$ and $K = 1$. We report the averaged accuracy of 500 tasks randomly sampled from $\mathcal{D}_{\text{target}}$.

4.4.2 Results

The main results are provided in Table 4.4 and Table 4.5, wherein CL-Reg, Cor-Reg, and CL-Reg + CorReg denote supervised pre-training regularized by the corresponding regularizer. We report the mean and standard deviation of our methods and IntentBERT variants. The following observations can be made. First, our proposed regularized supervised pre-training, with either CL-Reg or Cor-Reg, consistently outperforms all the baselines by a notable margin in most cases, indicating the effectiveness of our method. Our method also outperforms whitening transformation, demonstrating the superiority of the proposed joint fine-tuning and isotropization framework. Second, Cor-Reg slightly outperforms CL-Reg in most cases, showing the advantage of enforcing isotropy explicitly with the correlation matrix. Finally, CL-Reg and Cor-Reg show a complementary effect in many cases, especially on BANKING77. The above observations are consistent for both BERT and RoBERTa. It can be also seen that higher performance is often attained with RoBERTa.

Method	BANKING77	HINT3	HWU64
IntentBERT-ReImp	$.71_{(.04)}$	$.72_{(.03)}$	$.72_{(.03)}$
Supervised Pre-training+CL-Reg	$.77_{(.01)}$	$.78_{(.01)}$	$.75_{(.03)}$
Supervised Pre-training+Cor-Reg	$.79_{(.01)}$	$.76_{(.06)}$	$.80_{(.03)}$
Supervised Pre-training+CL-Reg+Cor-Reg	$.79_{(.01)}$	$.76_{(.05)}$	$.80_{(.02)}$

Table 4.6: The impact of the proposed regularizers on isotropy.

The observed improvement in performance comes with an improvement in isotropy. We report the change in isotropy based on BERT by the proposed regularizers in

Table 4.6. Both regularizers and their combination make the feature space more isotropic compared to IntentBERT-ReImp that only uses supervised pre-training. In addition, in general, Cor-Reg can achieve better isotropy than CL-Reg.

4.4.3 Analysis

Moderate isotropy is helpful. To investigate the relation between the isotropy of the feature space and the performance of few-shot intent detection, we tune the weight parameter λ of Cor-Reg to increase the isotropy and examine the performance. As shown in Fig. 4.5 (the results are obtained with BERT on 5-way 2-shot tasks), a common pattern is observed: the best performance is achieved when the isotropy is moderate. This observation indicates that it is important to find an appropriate trade-off between learning intent detection skills and learning an isotropic feature space. In our method, we select the appropriate λ by validation.

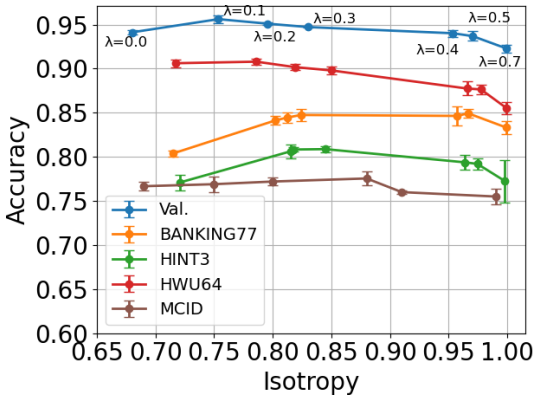


Figure 4.5: Relation between performance and isotropy.

Correlation matrix is better than covariance matrix as regularizer. In the design of Cor-Reg (Section 4.3.2), we use the correlation matrix, rather than the covariance matrix, to characterize isotropy, although the latter contains more information – variance. The reason is that it is difficult to determine the proper scale of the variances. Here, we conduct experiments using the covariance matrix, by

Method	BANKING77	Val.
Cov-Reg-1	82.19(.84)	94.52(.19)
Cov-Reg-0.5	82.62(.80)	94.52(.26)
Cov-Reg-mean	82.50(1.00)	93.82(.39)
Cor-Reg (ours)	**83.94(.45)**	**95.02(.22)**

Table 4.7: Comparison between covariance matrix and correlation matrix to implement the regularizer for isotropy.

pushing the non-diagonal elements (covariances) towards 0 and the diagonal elements (variances) towards 1, 0.5, or the mean value, which are denoted by Cov-Reg-1, Cov-Reg-0.5, and Cov-Reg-mean respectively in Table 4.7. It can be seen that all the variants perform worse than Cor-Reg. The experiment is conducted with BERT on 5-way 2-shot tasks.

Supervised Pre-training	CL-Reg	Cor-Reg	Batch Normalization	BANKING77
✓				80.38(.35)
✓			✓	82.38(.38)
✓	✓			83.45(.35)
✓	✓		✓	**84.18(.28)**
✓		✓		83.94(.45)
✓		✓	✓	**84.67(.51)**
✓	✓	✓		85.21(.58)
✓	✓	✓	✓	**85.64(.41)**

Table 4.8: The effect of combining batch normalization and our method.

Our method is complementary with batch normalization. Batch normalization [47] can potentially mitigate the anisotropy problem via normalizing each dimension with unit variance. We find that combining our method with batch normalization yields better performance, as shown in Table 4.8. The experiment is conducted with BERT and evaluated on 5-way 2-shot tasks. In addition, to confirm the effectiveness of the regularizers, we conduct ablation study, wherein we examine the effect of cross-

Cross-entropy	CL-Reg	Cor-Reg	BANKING77	Val.
✓	✓	✓	**85.21**(.58)	**95.41**(.25)
✓			80.38(.35)	93.62(.38)
	✓		77.66(.96)	88.74(.56)
		✓	76.17(2.04)	84.77(1.32)

Table 4.9: Ablation study.

entropy loss and the regularizer, respectively. As shown by Table 4.9, the optimal performance is achieved only when they are combined, because they have different functions: cross-entropy to deal with the task while the regularizer for isotropization.

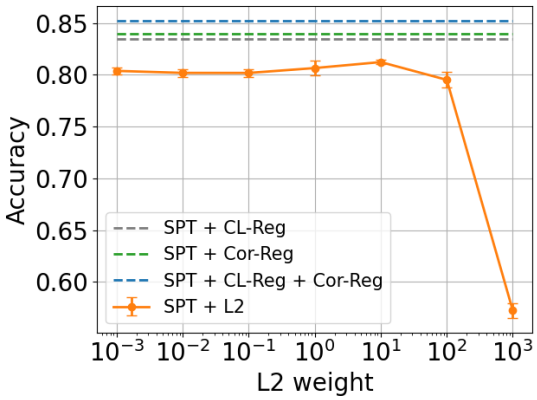


Figure 4.6: Comparison between our methods and L2 regularization. SPT denotes superivsed pre-training.

The performance gain is not from the reduction in model variance. Regularization techniques such as L1 regularization [94] and L2 regularization [42] are often used to improve model performance by reducing model variance. Here, we show that the performance gain of our method is ascribed to the improved isotropy (Table 4.6) rather than the reduction in model variance. To this end, we conduct experiments with BERT and 5-way 2-shot tasks on BANKING77, to compare our method against L2 regularization with a wide range of weights, and it is observed that reducing model

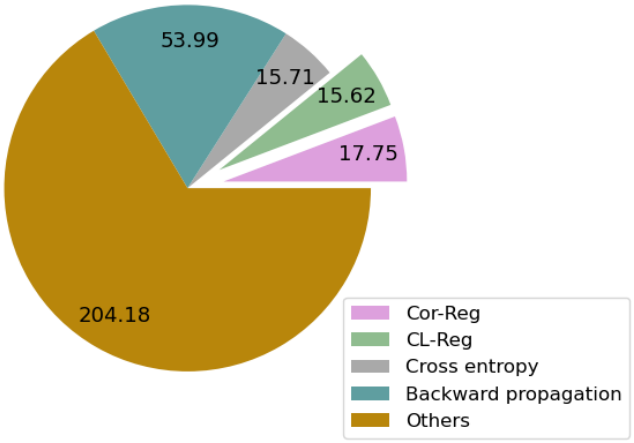


Figure 4.7: Run time decomposition of a single epoch. The unit is second.

variance cannot achieve comparable performance to our method, as shown in Fig. 4.6.

The computational overhead is small. To analyze the computational overheads incurred by CL-Reg and Cor-Reg, we decompose the duration of one epoch of our method using the two regularizers jointly. As shown in Fig. 4.7, the overheads of CL-Reg and Cor-Reg are small, only taking up a small portion of the time.

4.5 Conclusion

In this chapter, we have identified and analyzed the anisotropy of the feature space of a PLM fine-tuned on intent detection tasks. Further, we have proposed a joint training framework and designed two regularizers based on contrastive learning and correlation matrix respectively to increase the isotropy of the feature space during fine-tuning, which leads to notably improved performance on few-shot intent detection.

Chapter 5

Direct Fine-tuning PLMs for Data Efficiency

5.1 Motivation

The main obstacle for few-shot learning is commonly believed to be overfitting, i.e. the model trained with only a few examples tends to overfit to the training data and perform much worse on test data [98, 118]. To alleviate the problem, the mainstream approach is to transfer knowledge from *external resources* such as another labeled dataset, which has been widely used for few-shot image classification [26, 89] and few-shot intent detection [115, 32, 72].

Since recently emerged large-scale PLMs have achieved great success in various NLP tasks, most recent few-shot intent detection methods propose to fine-tune PLMs on external resources before applying them on the target task, which is known as *continual pre-training* [36, 114], as illustrated in Fig 5.1. The external resources utilized for continual pre-training include conversational corpus [103, 63, 99], natural language understanding datasets [121], public intent detection datasets (Chapter 3), and paraphrase corpus [61]. While these methods have achieved state-of-the-art results, the

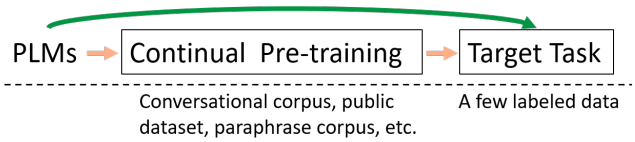


Figure 5.1: Illustration of continual pre-training (orange) and direct fine-tuning (green).

use of external training corpora induces extra data processing effort (e.g., SBERT-Paraphrase [61] uses 83 million sentence pairs from 12 datasets) as well as model bias (e.g., the trained model may be biased to the intent classes used in continual pre-training) [108, 105, 72].

It is commonly believed that directly fine-tuning PLMs with a small amount of data may generate unacceptable variance [53, 18]. However, it has been recently found that the instability may be caused by incorrect use of optimizer and insufficient training [70, 124]. Further, some studies [38, 56] have revealed that in sentiment analysis and paraphrase detection tasks, when directly fine-tuned with a small dataset, PLMs sush as BERT [17] demonstrate a certain level of resilience to overfitting.

Dataset	BANKING77	HINT3	HWU64	MCID
BERT Vocabulary	0.05	0.03	0.07	0.02
Generated Data	0.30	0.18	0.27	0.23

Table 5.1: Token overlap between generated data and test partitions of datasets.

In this chapter, we conduct a thorough investigation to explore the direct fine-tuning of PLMs for few-shot intent detection. Specifically, we take an empirical investigation into the overfitting issue when directly fine-tuning PLMs on few-shot intent detection tasks, which suggests that overfitting may not be a significant concern, since the test performance improves rapidly as the size of training data increases. Further, the model's performance does not degrade as training continues. It implies that early stopping is not necessary, which is often employed to prevent overfitting in few-shot

learning and requires an additional set of labeled data for validation. In addition, we find that direct fine-tuning (DFT) already yields decent results compared with continual pre-training methods. We further devise a DFT++ framework to fully exploit the given few labeled data and boost the performance.

Dataset	BANKING77	HINT3	HWU64	MCID
First half	79.19	60.65	79.69	79.67
Second half	88.93	75.44	82.62	84.19

(a) Model accuracy.

Dataset	BANKING77	HINT3	HWU64	MCID
Proportion	73.83%	51.18%	70.31%	70.84%

(b) The proportion of test data with features in both half-utterances.

Table 5.2: Half-utterance experiment results.

Using generative PLMs for data augmentation has been studied in NLP, but when it turns to intent detection with tens of closely relevant labels, label shift emerges as the obstacle [87]. We find that the generated data has contextual similarity to the test data, as indicated by the token overlap in Table. 5.1, wherein a notably higher token overlap is observed compared to the original vocabulary of the PLM. The novel data is generated by GPT-J with 5 data.

On the other hand, we find that utterances with the underlying intents are of a multi-view structure [1], i.e. there are multiple views (features) in the utterance indicating the intent. For instance, given the following utterance of label "travel alert",

how safe is visiting Canada this week,

both "safe" and "visiting" indicate the intent label. To verify such structure, we conduct a half-utterance experiment, wherein the classifier is trained with only the first-half utterance, or with the second-half. Take the above utterance as an example,

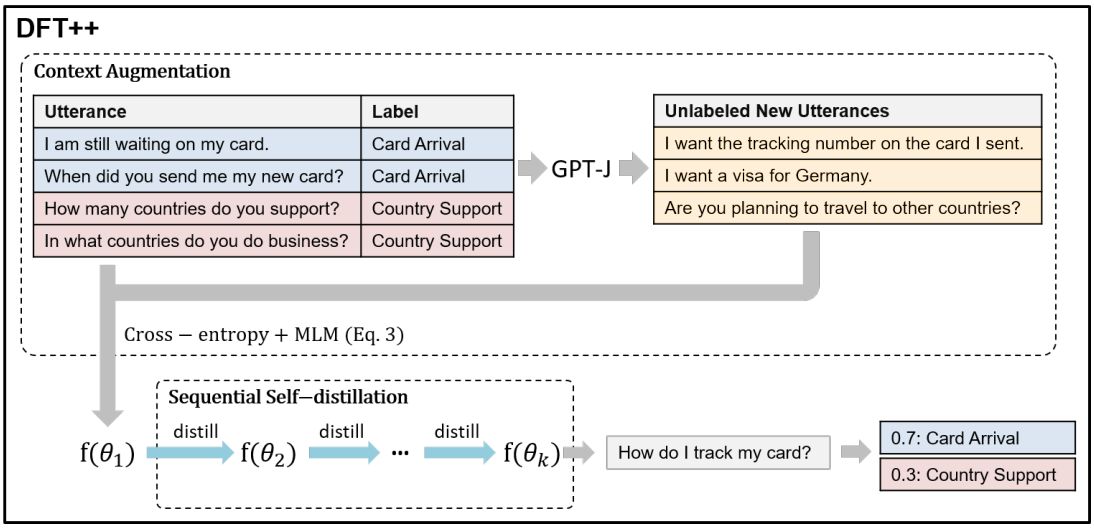


Figure 5.2: Illustration of DFT++ with 2 classes and 2 labeled examples per class.

the first half is "how safe is", while the second half, "visiting Canada this week". In this way, we tailor the training data into partial features, and then examine the model's performance. As shown by Table 5.2a, a highly discriminative classifier is trainable in both cases. Furthermore, across all datasets, a large proportion of the test data is observed that are distinguishable in both cases, with both models containing the first half feature and the second half, as shown in Table 5.2b. These experiments demonstrate the existence of multiple indicating features in the utterance. However, it is likely only one of them is learned by the model because one feature may suffice to discriminate the utterance from others, given a few training data.

According to the above two observations, DFT++ introduces a novel *context augmentation* mechanism by using a generative PLM to generate *contextually relevant unlabeled data* to enable better adaptation to target data distribution, as well as a sequential self-distillation mechanism to exploit the multi-view structure in data. A comprehensive evaluation shows that DFT++ outperforms state-of-the-art continual pre-training methods with only the few labeled data provided for the task, without resorting to external training corpora.

5.2 Direct Fine-tuning

We investigate a straightforward approach for few-shot intent detection – directly fine-tuning PLMs with the few-shot data at hand. However, it is a common belief that such a process may lead to severe overfitting. Before going into detail, we first formally define the problem.

5.2.1 Problem Definition

Few-shot intent detection aims to train an intent classifier with only a small labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_N$, where N is the dataset size, x_i denotes the i_{th} utterance, and y_i is the label. The number of samples per label is typically less than 10.

We follow the standard practice [92, 119] to apply a linear classifier on top of the utterance representations:

$$\mathrm{p}(y|\mathbf{h}_i) = \text{softmax}\left(\mathbf{W}\mathbf{h}_i + \mathbf{b}\right) \in \mathbb{R}^L, \tag{5.1}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the representation of the i_{th} utterance in $\mathcal{D}$, $\mathbf{W} \in \mathbb{R}^{L\times d}$ and $\mathbf{b} \in \mathbb{R}^L$ are the parameters of the linear layer, and L is the number of classes. We use the representation of the [CLS] token as the utterance embedding $\mathbf{h}_i$. The model parameters $\theta = \{\phi, \mathbf{W}, \mathbf{b}\}$, with ϕ being the parameters of the PLM, are trained on $\mathcal{D}$. We use a cross-entropy loss $\mathcal{L}_{\text{ce}}(\cdot)$ to learn the model parameters:

$$\theta = \arg\min_{\theta} \mathcal{L}_{\text{ce}}\left(\mathcal{D}; \theta\right). \tag{5.2}$$

Unlike the popular approach of continual pre-training [121, 118, 120], DFT fine-tunes PLMs directly on the few-shot data, which may experience overfitting, leading to sub-optimal performance. To examine this issue, we conduct the following experiments.

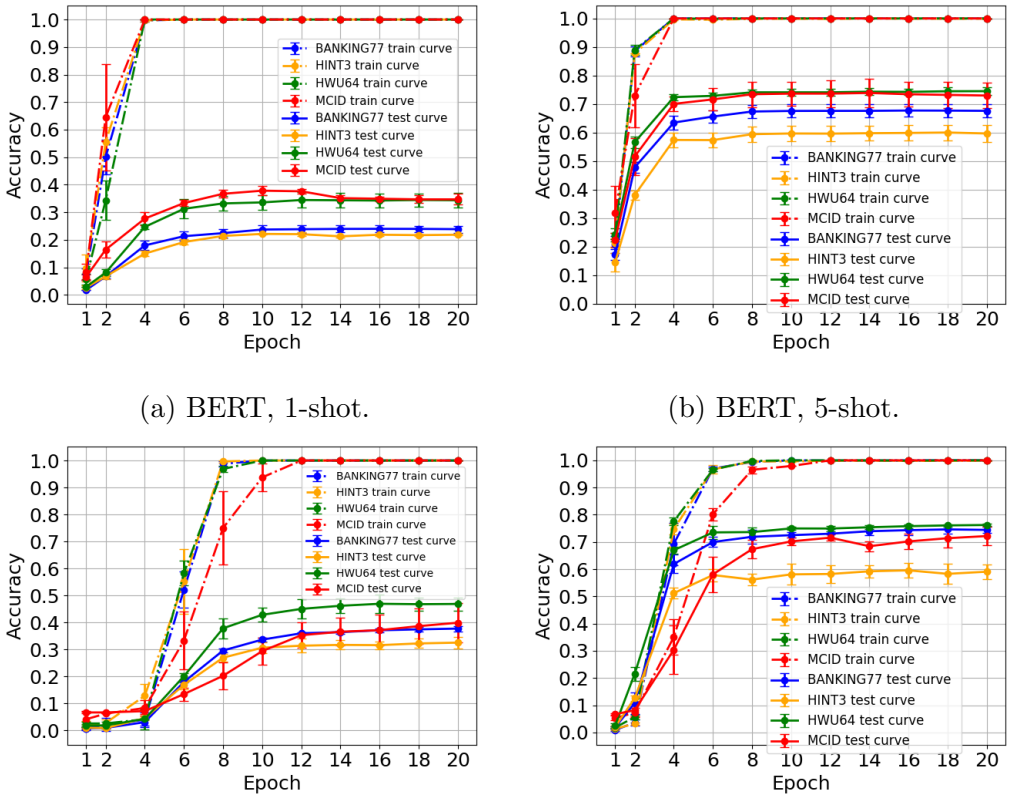
5.2.2 Experiments

Datasets We utilize four datasets for evaluation: **HINT3**, **BANKING77**, **MCID** and **HWU64**. To simulate few-shot scenarios, we randomly sample K samples per label from the training set of each dataset to form the dataset $\mathcal{D}$.

Baselines To evaluate DFT, we compare it against IsoIntentBERT [118], a competitive baseline applying continual pre-training with public intent detection datasets. We follow the original work to pre-train BERT on OOS [52], a multi-domain public intent detection dataset containing diverse semantics, and then perform in-task fine-tuning on the small dataset $\mathcal{D}$.

Results and Findings We plot the learning curves of DFT in Fig. 5.3, and the following observations can be drawn. First, comparing the results in 1-shot and 5-shot scenarios, the test performance of DFT improves drastically as the number of labeled examples rises from 1 to 5, leading to a fast reduction in the performance gap between the training and test performance. Second, the test performance does not deteriorate as the training progresses, and the learning curves exhibit a flat trend. These observations are consistent across a wide spectrum of datasets and different models (BERT and RoBERTa), including both 1-shot and 5-shot scenarios. The observations also align with previous findings in sentiment analysis [56] and paraphrase detection [38] tasks.

The flat learning curves indicate that early stopping is not necessary, which is often used to prevent overfitting and requires an additional set of labeled data. This is important for practitioners because model selection has been identified as a roadblock for *true few-shot learning* [76], where the labeled data is so limited that it is not worth setting aside a portion of it for early stopping. On the other hand, as shown by Fig. 5.4, the benefit from continued pre-training decays quickly, i.e. the performance



(a) BERT, 1-shot.

(b) BERT, 5-shot.

(c) RoBERTa, 1-shot.

(d) RoBERTa, 5-shot.

Figure 5.3: Training and test learning curves of DFT with BERT and RoBERTa as text encoder respectively.

gap between DFT and IsoIntentBERT reduces rapidly, which casts doubt on the necessity of continual pre-training. Thus, we raise an intriguing question:

- With only the given few labeled data, is it possible to achieve comparable or better performance than continual pre-training methods?

Our attempt to answer the question leads to DFT++, a framework designed to fully exploit the given few labeled data, which provides an affirmative answer.

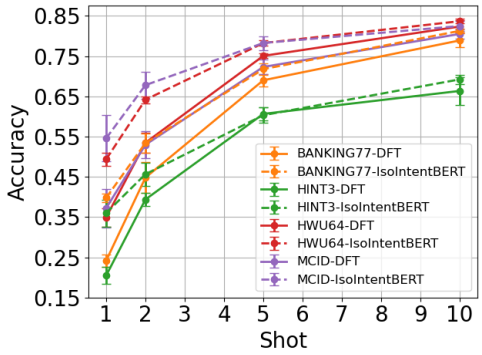


Figure 5.4: Comparison between DFT (solid lines) and IsoIntentBERT (dashed lines).

5.3 Push the Limit of Direct Fine-Tuning

To push the limit of few-shot intent detection with only a few labeled data at hand and without using any external training corpora, DFT++ introduces two mechanisms, as shown in Fig. 5.2. The first is a novel context augmentation mechanism, wherein the few data are used to prompt GPT-J, a generative PLM, to generate contextually relevant unlabeled utterances to better model target data distribution. The second is a sequential self-distillation mechanism further boosting the performance.

5.3.1 Context Augmentation

Unlike continual pre-training methods that leverage external training corpora, we use the few data to solicit knowledge from generative PLMs. An intuitive way is data augmentation, which prompts the model to generate new utterances with the given intent class. However, as suggested by [87], data augmentation for intent detection with tens of intent classes is challenging. Hence, we propose to exploit contextual relevance in an unsupervised manner instead. Specifically, for each intent class, we compose the few data into a prompt and then feed it to GPT-J [100], a powerful generative PLM, to generate novel unlabeled utterances. Fig. 5.5 gives an example of the prompt and generated results in a 5-shot scenario, wherein the green utterances

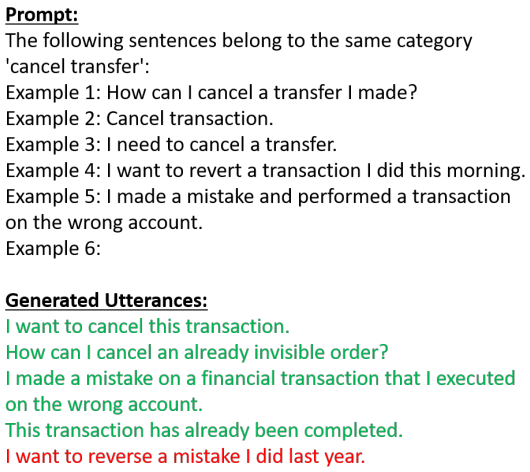


Figure 5.5: An example of the prompt and generated utterances.

are successful cases, while the red one is a failure case. The generated unlabeled data is combined with the given utterances in $\mathcal{D}$ to compose a corpus $\mathcal{D}_{\text{aug}} = \{x_i\}_i$, which can be used for masked language modeling (MLM). Hence, the model parameters θ are learned by simultaneously minimizing both the cross-entropy loss $\mathcal{L}_{\text{ce}}$ and the MLM loss $\mathcal{L}_{\text{mlm}}$:

$$\theta = \arg\min_{\theta} \left(\mathcal{L}_{\text{ce}}(\mathcal{D}; \theta) + \lambda \mathcal{L}_{\text{mlm}}(\mathcal{D}_{\text{aug}}; \theta) \right), \tag{5.3}$$

where λ is a balancing parameter.

Notice that there is a critical difference between the proposed context augmentation and conventional data augmentation methods. Context augmentation generates contextually relevant data (i.e., utterances with similar context to the given input but not necessarily belong to the same label class), and we use the generated data in an unsupervised manner via MLM. In contrast, conventional data augmentation methods generate new utterances with the same label as the given utterance and utilize them in a supervised manner.

5.3.2 Sequential Self-distillation

To further boost performance, we employ self-distillation [69, 1] (Fig. 5.2). The knowledge in the learned model is distilled into another model with the same architecture by matching their output logits[1]:

$$\theta_k = \arg\min_{\theta_k} \mathrm{KL}\left(\frac{\mathrm{f}(\mathcal{D};\theta_k)}{t}, \frac{\mathrm{f}(\mathcal{D};\theta_{k-1})}{t}\right), \tag{5.4}$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler (KL) divergence, $\mathrm{f}(\cdot)$ is the output logit of the model, and t is the temperature parameter. We adopt the born-again strategy [29] to iteratively distill the model into a sequence of generations. Hence, the model at k_{th} generation with parameters θ_k is distilled to match the $(k-1)_{\text{th}}$ generation with parameters θ_{k-1}. Self-distillation can provably improve model performance if the data has a multi-view structure, i.e., the data has multiple features (views) to help identify its class [1]. Such structures naturally exist in utterances (Section 5.1). Sequential self-distillation can help to learn all features, as shown in [1].

5.4 Experiments

5.4.1 Setup

We evaluate DFT++ on the same benchmarks used to evaluate DFT. We compare DFT++ with state-of-the-art continual pre-training methods. Since early stopping is not necessary, as demonstrated in the section 5.2.2, we combine the validation and test partitions for a more comprehensive evaluation.

Baselines. We compare the proposed method against the following baselines. **TOD-BERT** [103] conducts continual pre-training on dialogue corpus with MLM and response objectives. **DNNC-NLI** [121] and **SE-NLI** [61] employ NLI datasets.

[1]We have also tried to add a cross-entropy term [93], but find it hurts the performance.

DNNC-NLI is equipped with a BERT-style pair-wise similarity model and a nearest neighbor classifier. SE-NLI employs sentence encoder [83] with siamese and triplet architecture to learn the semantic similarity. **DNNC-Intent**, **CPFT** [120], **IntentBERT** [119] and **IsoIntentBERT** [118] use external intent detection datasets. DNNC-Intent shares the same model structure as DNNC-NLI. CPFT adopts contrastive learning and MLM. IntentBERT employs standard supervised pre-training, based on which IsoIntentBERT introduces isotropization to further improve model performance. **SE-Paraphrase** [61] exploits a paraphrase corpus, using the same model architecture for sentence encoding as SE-NLI. **One-to-All** [22] is the most recent work that encodes the entire intent space together with the query utterance for more accuracy classification.

For all the baselines, we download the publicly released model if available. Otherwise, we follow the original work's guidelines to perform continual pre-training. Next, we perform standard fine-tuning similar to DFT, using hyperparameters searched within the same range as our method, with three exceptions: DNNC-NLI, DNNC-Intent, and CPFT. For these methods, we use the original design and training configuration for in-task fine-tuning.

In addition, we compare DFT++ against **CINS** [66], the most recent prompt-based method. CINS addresses intent detection by converting it into a cloze-filling problem through a carefully designed prompt template. Similar to our method, CINS directly fine-tunes PLMs on a limited amount of data.

Our method. We evaluate our method and the baselines based on two popular PLMs: BERT [17] and RoBERTa [60]. The representation of the token [CLS] is used as the utterance embedding. For a fair comparison, we select the hyper-parameters with the same validation data as used by the baselines, i.e., we follow IsoIntentBERT to use a portion of OOS dataset as the validation data. The best hyper-parameters and grid search range are given in the appendix.

Method	BANKING77	HINT3	HWU64	MCID
TOD-BERT	67.69(1.37)	56.33(2.14)	74.83(1.11)	66.37(2.65)
DNNC-NLI	68.48(1.15)	59.05(1.02)	72.25(1.39)	67.35(2.09)
DNNC-Intent	70.36(1.85)	58.08(4.98)	69.86(4.27)	70.80(3.16)
CPFT	70.96(2.45)	61.63(2.64)	73.63(1.74)	71.54(4.97)
IntentBERT	70.64(1.02)	58.96(1.50)	**77.60(.31)**	76.67(.84)
IsoIntentBERT	71.78(1.40)	60.33(1.95)	**78.26(.69)**	**78.28(1.72)**
SE-Paraphrase	**71.92(.84)**	**62.28(.77)**	76.75(.63)	**78.32(2.12)**
SE-NLI	70.03(1.47)	**61.69(1.59)**	75.10(1.17)	74.54(1.86)
DFT	69.01(1.54)	60.65(1.60)	75.07(.53)	72.32(1.80)
DFT++ (w/ CA)	**72.23(1.80)**	60.53(2.73)	76.73(1.05)	77.45(1.66)
DFT++ (w/ SSD)	68.86(1.49)	61.51(1.88)	75.05(1.36)	74.17(1.09)
DFT++ (w/ CA, SSD)	**72.90(.89)**	**63.08(1.17)**	**77.73(1.02)**	**79.43(.84)**

(a) 5-shot evaluation results.

Method	BANKING77	HINT3	HWU64	MCID
TOD-BERT	79.71(0.91)	66.42(2.19)	82.15(0.47)	74.66(1.52)
DNNC-NLI	74.53(4.83)	65.12(1.96)	77.91(1.11)	75.20(1.28)
DNNC-Intent	78.85(1.56)	64.56(3.64)	74.87(3.02)	78.60(1.49)
CPFT	79.44(.80)	69.85(1.21)	80.59(.61)	79.38(1.60)
IntentBERT	81.18(.34)	68.96(1.50)	**83.55(.21)**	81.60(1.41)
IsoIntentBERT	**81.30(.50)**	69.23(1.16)	**83.70(.59)**	**82.51(1.23)**
SE-Paraphrase	81.18(.33)	**70.00(1.01)**	82.88(.48)	**83.08(1.32)**
SE-NLI	80.58(1.13)	68.37(1.55)	82.57(.79)	81.20(1.80)
DFT	78.92(1.69)	66.36(3.48)	82.38(1.49)	80.53(1.15)
DFT++ (w/ CA)	**82.33(.72)**	**70.36(1.90)**	82.61(.23)	81.27(1.41)
DFT++ (w/ SSD)	80.32(.81)	68.82(2.49)	82.14(.92)	81.44(1.08)
DFT++ (w/ CA, SSD)	**82.66(.50)**	**70.47(2.56)**	**83.45(.38)**	**82.83(.76)**

(b) 10-shot evaluation results.

Table 5.3: Evaluation of DFT++ based on BERT.

Implementation details. We use Python, PyTorch library and Hugging Face library to implement the model. We adopt *bert-base-uncased* and *roberta-base* with

Method	BANKING77	HINT3	HWU64	MCID
DNNC-NLI	73.90(1.27)	59.73(0.89)	73.06(1.70)	63.74(3.79)
DNNC-Intent	72.97(1.46)	61.15(1.74)	69.74(1.85)	72.44(2.50)
CPFT	70.94(1.08)	58.17(3.44)	74.36(1.15)	78.20(1.72)
IntentRoBERTa	75.23(.89)	60.77(1.60)	78.97(1.26)	77.25(2.05)
IsoIntentRoBERTa	75.05(1.92)	59.79(2.72)	78.09(1.06)	78.40(2.03)
SE-Paraphrase	76.03(.64)	**63.96(.02)**	76.50(.45)	**80.78(1.36)**
SE-NLI	76.56(.69)	62.60(2.45)	78.53(.84)	**79.43(3.17)**
One-to-All¶	**79.75(.78)**	-	**79.89(.30)**	-
DFT	76.11(1.16)	61.39(1.51)	76.72(.94)	76.39(1.18)
DFT++ (w/ CA)	**78.74(1.00)**	**63.17(2.20)**	**79.02(.89)**	76.51(2.77)
DFT++ (w/ SSD)	76.25(1.67)	61.30(2.31)	77.57(.62)	78.73(2.30)
DFT++ (w/ CA, SSD)	**78.90(.50)**	**63.61(1.80)**	**79.93(.92)**	**80.16(2.74)**

(a) 5-shot evaluation results.

Method	BANKING77	HINT3	HWU64	MCID
DNNC-NLI	79.51(2.56)	64.05(2.30)	78.12(1.86)	73.72(1.82)
DNNC-Intent	77.69(5.06)	66.45(1.06)	72.30(3.61)	78.64(1.69)
CPFT	78.57(.75)	61.07(2.37)	79.46(.81)	83.04(1.74)
IntentRoBERTa	83.94(.33)	68.91(1.24)	84.26(.84)	82.67(1.43)
IsoIntentRoBERTa	84.49(.43)	69.08(1.59)	84.15(.58)	83.20(1.89)
SE-Paraphrase	82.85(.89)	69.14(2.08)	81.25(.97)	83.12(.86)
SE-NLI	84.65(.26)	69.91(1.82)	84.81(.45)	**84.13(1.25)**
DFT	84.77(.43)	68.40(1.21)	84.00(.34)	82.55(1.15)
DFT++ (w/ CA)	**85.95(.34)**	**71.30(1.54)**	**85.49(.35)**	**83.98**1.17)
DFT++ (w/ SSD)	**84.95(.53)**	**70.12(1.35)**	**84.91(.45)**	83.371.64)
DFT++ (w/ CA, SSD)	**86.14(.19)**	**71.80(1.88)**	**86.21(.28)**	**84.80(.79)**

(b) 10-shot evaluation results.

Table 5.4: Evaluation of DFT++ based on RoBERTa. ¶ denotes results from [22].

around 110 million parameters. We use AdamW as the optimizer. We use different learning rates for PLMs and the linear classifier, determined by grid-search. The parameter for weight decay is set to $1e-3$. We employ a linear scheduler with the

warm-up proportion of 5%. We fine-tune the model for 200 epochs to guarantee convergence. The experiments are conducted with Nvidia RTX 3090 GPUs. We repeat all experiments for 5 times, reporting the averaged accuracy and standard deviation.

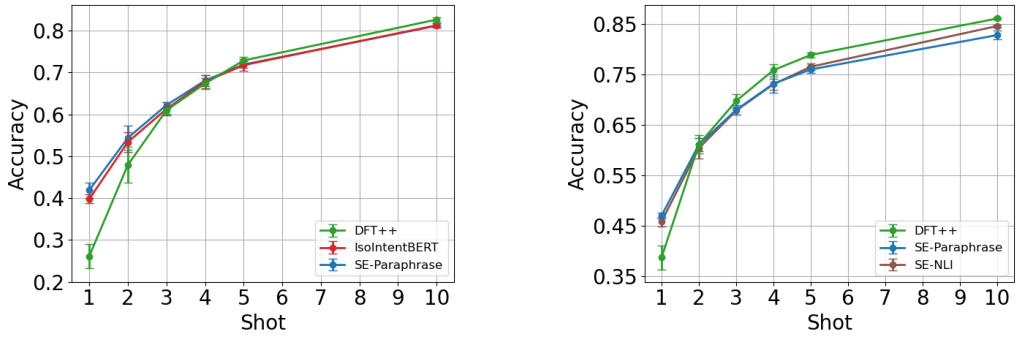
5.4.2 Results

We first examine the performance using a moderately small amount of data, specifically 5-shot and 10-shot scenarios. The results are summarized in Table 5.3 and Table 5.4. Remarkably, DFT++ performs comparably to a diverse set of baselines that leverage external resources, despite the fact that it solely utilizes the limited few-shot data available. The superiority of DFT++ can be attributed to the effective utilization of context augmentation and sequential self-distillation, both of which demonstrate improved results when applied independently in most cases. Similar phenomenon is observed when using the stronger base model RoBERTa, as shown in Table 5.4. In these tables, CA denotes context augmentation, SSD denotes sequential self-distillation, and we highlight the top 3 results. Moreover, as shown in Table 5.5, in most cases, DFT++ also outperforms CINS, the most recent prompt-based method, especially when RoBERTa is employed, despite that CINS employs T5-base [78] with 220 million parameters, which is almost twice the size of our base model. In the tables, we report the mean value and standard deviation.

5-shot	Bank	Home	Utility	Auto
CINS¶	89.1	80.2	95.4	**93.7**
DFT++ (BERT)	**91.39**(.78)	**82.11**(4.09)	**96.16**(.41)	90.64(.93)
DFT++ (RoBERTa)	**93.76**(.46)	**86.21**(2.94)	**97.39**(.50)	**93.31**(1.21)

Table 5.5: The comparison of DFT++ against CINS. ¶ denotes results from [66]. The top 2 results are highlighted.

To study the impact of the number of labeled data on performance, we reduce the number to only 1 sample per label and present the results in Fig. 5.6. We experiment



(a) BERT-based experiments.

(b) RoBERTa-based experiments.

Figure 5.6: The impact of the size of labeled data on performance.

with BANKING77, a challenging fine-grained dataset and compare DFT++ with the top 2 baselines. When using BERT, we observe that DFT++ begins to outperform the baselines at a crossing point of 4. When using RoBERTa, the crossing point is even smaller, at 2, which is quite surprising. We have also observed similar phenomena on other datasets, as detailed in the appendix. The observations confirm our claim that the overfitting issue in directly fine-tuning PLMs for few-shot intent detection may not be as severe as initially presumed. The performance disadvantage due to overfitting can be effectively alleviated by leveraging other techniques to exploit the limited available data, even without resorting to the continual pre-training approach. However, in scenarios with an extremely small number of labeled data, the transferred knowledge from continual pre-training still provides significantly better performance compared to DFT++.

5.4.3 Analysis

Comparison between contextual augmentation and conventional data augmentation methods. We compare our proposed context augmentation with the following conventional data augmentation methods. Easy Data Augmentation (EDA) [102] modifies a small number of utterances, e.g., through word swapping, to

Method	BANKING77	HINT3	HWU64	MCID
DFT	69.01(1.54)	60.65(1.60)	75.07(.53)	72.32(1.80)
EDA	68.81(1.97)	60.50(3.06)	74.68(.81)	73.10(.64)
BT	69.65(1.39)	60.50(1.40)	74.15(.84)	75.15(2.04)
PromptDA	71.62(.72)	**61.51(2.20)**	76.59(.89)	77.16(.98)
SuperGen	64.83(1.06)	57.30(1.41)	69.52(0.56)	72.55(1.37)
GPT-J-DA	71.84(1.41)	60.24(.83)	70.72(.78)	73.92(2.77)
Contextual Augmentation	**72.23(1.80)**	60.53(2.73)	**76.73(1.05)**	**77.45(1.66)**

(a) BERT-based evaluation results.

Method	BANKING77	HINT3	HWU64	MCID
DFT	76.11(1.16)	61.39(1.51)	76.72(.94)	76.39(1.18)
EDA	74.74(1.08)	62.04(2.49)	75.88(1.59)	77.17(1.85)
BT	75.12(1.03)	60.83(1.16)	77.31(.72)	77.49(2.71)
PromptDA	76.56(1.15)	60.56(1.37)	77.57(1.12)	77.60(1.94)
SuperGen	70.42(0.19)	57.64(1.33)	71.28(0.78)	73.99(1.79)
GPT-J-DA	76.58(1.30)	62.16(1.83)	76.59(.94)	**77.91(2.22)**
Contextual Augmentation	**78.74(1.00)**	**63.17(2.20)**	**79.02(.89)**	76.51(2.77)

(b) RoBERTa-based evaluation results.

Table 5.6: Comparison of our proposed contextual augmentation against conventional data augmentation methods.

generate new augmented instances. Back-translation (BT) [23] translates an utterance into another language and then translates it back[2]. PromDA [101] and SuperGen [65] are recent data augmentation methods leveraging generative PLMs. GPT-J-DA [87] exploits the data generated by GPT-J in a supervised manner. The results in Table 5.6 show context augmentation is more robust against data shift. Note that SuperGen is designed for coarse-grained tasks with only two or three labels, such as

[2]We use French as the intermediate language, and utilize T5-base [78] and opus-mt-fr-en [95] for translation.

Input	Good	Bad
"Is there a reason why my card was declined when I attempted to withdraw money?", "How come I can not get money at the ATM?", "Why can not I withdraw cash from this ATM?", "Why will not the ATM give me cash?", "This morning, I wanted to make a withdrawal before work but my card was declined, please double check it for me as this is the first time it was declined."	"ATM will not let me withdraw my money my card as refused please help", "I withdrew less than I expected from the ATM on monday", "My wallet was stolen but my ATM card was within safely", "I spent a fortune last week and have none left on my card can you reverse refund the fees","Please give me the code that I can use in the ATM for my face to use my card"	"Why did my card never get a their villages and journey?", "An autofill took place but there was nothing to approve.", "Can I get one form my card after I have made a ctifre?", "Family needs money for the holidays they said they can not make it I hope you can help even if it is not much."
"Please order take from Jasons Deli.", "Can you please order some food for me?", "Can you look up Chinese takeout near here?", "Can i order takeaway from Spanish place?", "Find and order rasgulla of janta sweet home pvt ltd."	"I need to get some gluten free cookies for my daughter", "Can you do ticket counter take away", "How can I order Chinese food", "Delivery service please order some takeaway jahdi", "Order beef kasundi bewa rasgulla and dosa will be ready in 10 mins"	"Please make some reservation if you want booking on myhotelcom", "Drive take from a taxi", "Warehouse 26723", "Please make some reservation if you want booking on myhotelcom"

Table 5.7: Utterances generated by GPT-J. The first row corresponds to the label "Declined Cash Withdrawal" from BANKING77. The second row corresponds to the label "Takeaway Order" from HWU64. Good examples exhibit semantic relevance to the input data, while bad examples are irrelevant. Green words are highlighted to indicate semantic relevance, while the underlined words deviating the sentence from the original label.

sentiment classification. As a result, it may not scale effectively to intent detection tasks that involve a larger number of intents, typically ranging in the tens. The comparison between context augmentation and GPT-J-DA highlights the superiority of unsupervised exploitation of the generated data. The inconsistent effectiveness of GPT-J-DA is also reported by [87]. The experiment is conducted with 5-shot tasks. The best results are highlighted.

Quality of context augmentation. To demonstrate the quality of the data generated by context augmentation, we provide some good and bad examples of generated utterances in Table 5.7. It is observed that GPT-J is able to generate grammatically fluent utterances that exhibit a high level of contextual relevance to the input utter-

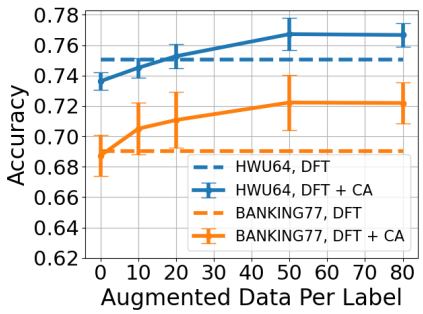
ances, which are utilized by DFT++ to better model the target data distribution. On the other hand, as also observed in [87], some of the generated utterances deviate from the original label and, therefore, are not suitable for data augmentation. However, DFT++ works around this issue by focusing solely on leveraging contextual relevance, resulting in improved robustness against data shift (Table 5.6).

IsoIntentBERT	DFT++	BANKING77	HWU64
✓		71.78(1.40)	78.26(.69)
✓	✓	**73.53(1.33)**	**80.20(1.20)**
SE-Paraphrase	DFT++	BANKING77	HWU64
✓		71.92(.84)	76.75(.63)
✓	✓	**73.21(1.24)**	**78.34(.31)**

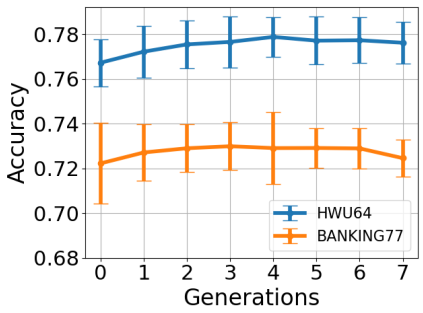
Table 5.8: Complementarity of DFT++ and continued pre-training with experiments conducted on 5-shot tasks.

Complementarity of continual pre-training and DFT++. Continual pre-training and DFT++ mitigate overfitting from different aspects. The former leverages external data, while the latter maximizes the utilization of the limited available data. Hence, it is likely that they are complementary. To support this claim, we present empirical results demonstrating their complementarity in Table 5.8. It is observed that when combined with DFT++, the two competitive methods, IsoIntentBERT and SE-Paraphrase, are both benefited.

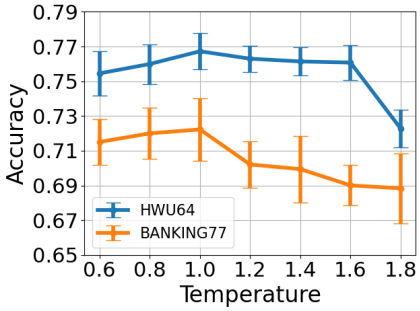
Impact of hyper-parameters. We study the impact of several key hyper-parameters, including the size of the generated data, the number of self-distillation generations, the temperature of GPT-J and self-distillation. The experiments are conducted in 5-shot scenarios. As visualized in Fig. 5.7a, a positive correlation is found between the performance and the size of the augmented data. The performance saturates after the data size per label reaches 50. It is noted that when only the given data are used for MLM, i.e., when the generated data size is 0, MLM has an adversarial effect probably due to overfitting on the few given data. Such negative effect is successfully



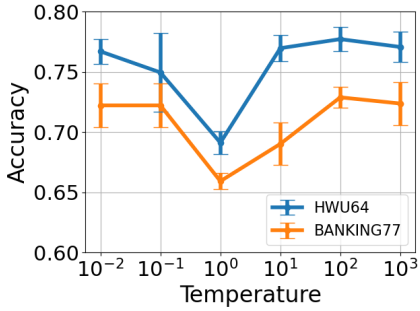
(a) Augmented data size.



(b) Distillation generations.



(c) GPT-J temperature.



(d) Self-distillation temperature.

Figure 5.7: Impact of hyper-parameters. CA denotes context augmentation.

alleviated by context augmentation. As for self-distillation generations (Table 5.7b), we find that multiple generations of self-distillation are necessary to achieve better performance. We show the impact of the temperature parameters in Fig. 5.7. The temperature parameter of GPT-J controls the diversity of the generated context. A higher temperature makes the generated text more diverse. As shown in the figure, the best performance is reached when the diversity is moderate. For self-distillation, both small and large temperatures produce good results.

Comparison with alternative context augmentation methods. We have also studied alternative context augmentation methods. The first one is Easy Data Augmentation (EDA) [102] with random synonym replacement, insertion, swap, and deletion. The second approach involves manually collecting a domain-specific corpus. We

Method	BANKING77	
	5-shot	10-shot
DFT	69.01(1.54)	78.92(1.69)
DFT + External	67.84(.82)	81.23(.66)
DFT + EDA	70.61(1.78)	81.83(.41)
DFT + GPT-J	**72.22(1.80)**	**82.33(.72)**

Table 5.9: The comparison of our proposed GPT-J-based context augmentation with other alternatives. "External" denotes Wikipedia corpus collection.

conduct experiments on BANKING77, since it focuses on a single domain, making it convenient to collect the corpus. We extract web pages from Wikipedia[3] with keywords that are closely relevant to "Banking", such as "Bank" and "Credit card". The keywords can be found in the appendix. As shown by Table 5.9, our GPT-J-based context augmentation outperforms the alternatives. We attribute the superiority to the grammatical fluency achieved by leveraging the generative power of GPT-J, which is typically compromised by EDA. Additionally, the high degree of semantic relevance observed in our approach is rarely guaranteed in the noisy corpus collected from Wikipedia.

5.5 Conclusion

In this chapter, we compare two approaches: direct fine-tuning and continual pre-training. We show that the overfitting issue may not be as significant as commonly believed. In most cases, our proposed framework, DFT++, demonstrates superior performance compared to mainstream continual pre-training methods that rely on external training corpora, indicating that the continual pre-training stage can be removed to improve data-efficiency.

[3]https://en.wikipedia.org

5.6 Appendix

Hyper-parameters. We determine the hyper-parameters by grid search. The best hyper-parameters and the search range are summarized in Table 5.10 and Table 5.11, respectively. lr_{PLM} and lr_{cls} denote the learning rate of the PLM and the linear classifier, respectively. context_size is the size of the augmented contextual utterances per label. iteration is the number of iterations/generations in sequential self-distillation. The grid search is performed with OOS dataset. Specifically, we follow Chapter 4 to use the two domains "Travel" and "Kitchen dining" as the validation set. To guarantee a fair comparison, the same validation set is also employed for all the baselines.

PLM	Hyper-parameter
BERT	$\text{lr}_{\text{PLM}} = 2e-4$, $\text{lr}_{\text{cls}} = 2e-5$, $\lambda = 1.0$, context_size =50, $t = 100$, iteration=6.
RoBERTa	$\text{lr}_{\text{PLM}} = 2e-5$, $\text{lr}_{\text{cls}} = 2e-3$, $\lambda = 0.1$, context_size =50, $t = 40$, iteration=5.

Table 5.10: Hyper-parameters of DFT++.

Parameter	Range
lr_{PLM}	$\{2e-5, 2e-4, 2e-3\}$
lr_{cls}	$\{2e-5, 2e-4, 2e-3\}$
λ	$\{0.01, 0.1, 1.0, 10.0\}$
context_size	$\{1, 2, 5, 10, 20, 50, 80\}$
t	$\{0.1, 1, 10, 40, 80, 100, 200, 500\}$
iteration	$\{1, 2, 3, 4, 5, 6, 7\}$

Table 5.11: Grid search range of hyper-parameters.

Impact of the number of labeled data on performance. We provide the full results in Fig. 5.8. It is observed that DFT++ outperforms many competitive

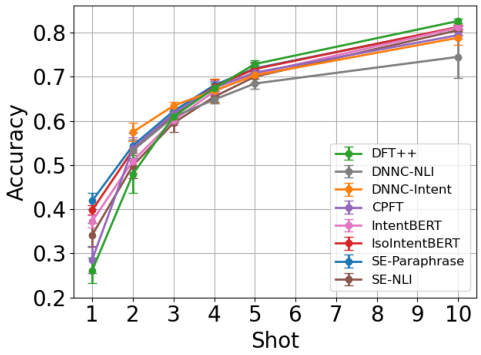
methods fine-tuned on extra data even when the number of labeled data is small.

"Bank", "Credit", "Debt", "Payment", "Fund", "Credit card", "Banking agent", "Bank regulation", "Cheque", "Coin", "Deposit account", "Electronic funds transfer", "Finance", "Internet banking", "Investment banking", "Money", "Wire transfer", "Central bank", "Credit union", "Public bank", "Cash", "Call report", "Ethical banking", "Loan", "Mobile banking", "Money laundering", "Narrow banking", "Private banking"

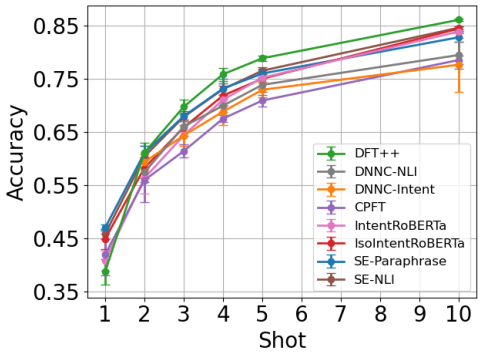
Table 5.12: Key words used to collect the corpus from Wikipedia.

Keywords used to collect the corpus for an alternative context augmentation method. As introduced in section 5.4.3, one alternative context augmentation method involves manually collecting a domain-specific corpus. We experiment with BANKING77. To collect an external corpus, we extract web pages from Wikipedia[4] with keywords closely related to "Banking", such as "Bank" and "Credit card". The adopted keywords are summarized in Table 5.12.

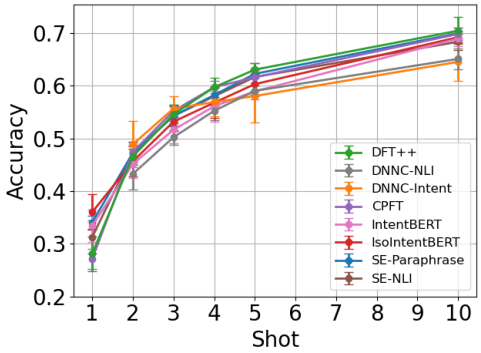
[4]https://en.wikipedia.org



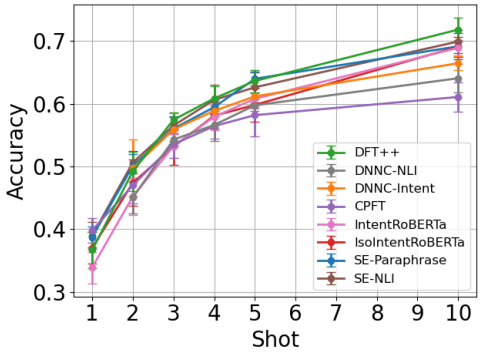
(a) BERT, BANKING77.



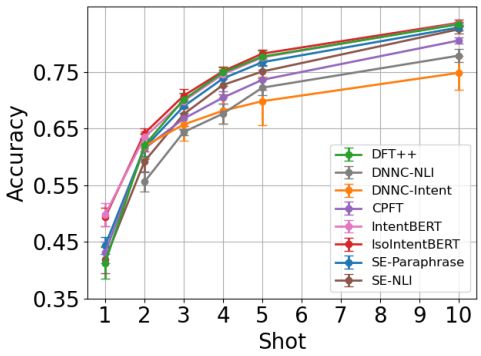
(b) RoBERTa, BANKING77.



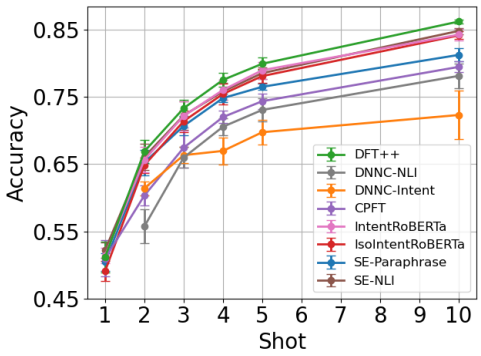
(c) BERT, HINT3.



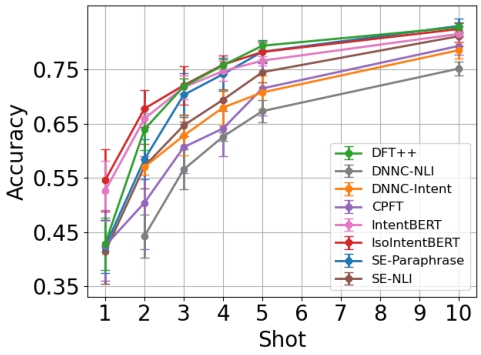
(d) RoBERTa, HINT3.



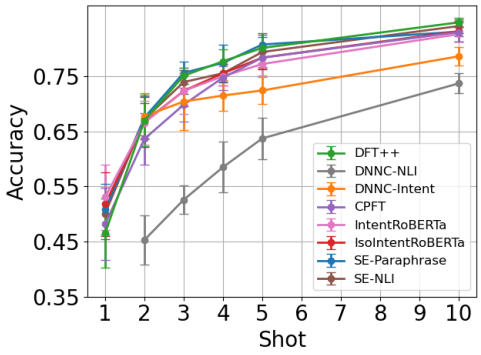
(e) BERT, HWU64.



(f) RoBERTa, HWU64.



(g) BERT, MCID.



(h) RoBERTa, MCID

Figure 5.8: Impact of the number of labeled data on model performance.

Chapter 6

Compression Method for Model Efficiency

6.1 Motivation

In previous chapters, we have achieved remarkable efficacy and data-efficiency, but PLMs incur significant computational overheads stemming from the gigantic model size, usually containing millions or billions of parameters [17, 78, 28]. Therefore, the inference of PLMs usually requires high-performance processors, memory capacities, and power consumption, which poses challenges when deploying the model on resource-constrained devices such as edge devices and mobile devices [110]. However, there are a few works to develop a small intent detector under few-shot scenario. To our best knowledge, attempts towards this objective is restricted to [88], which assumes the access to an external set of annotated intent detection data, but we focus on a more challenging scenario that no external data is available.

To develop a small intent detector, two challenges have to be tackled. The first one is *neural model compression* (excluding the vocabulary) under the few-shot constraint. To tackle this issue, [64] adopt a generative PLM to augment the data required for

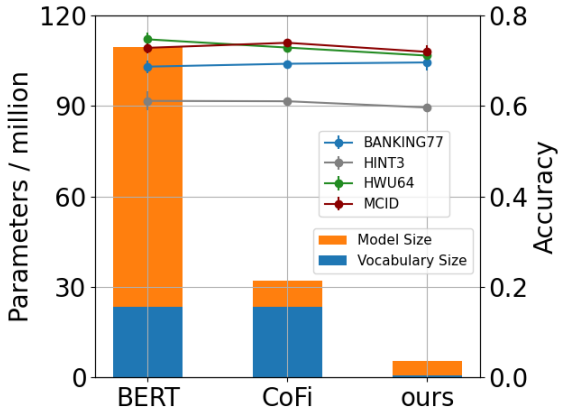


Figure 6.1: The efficacy of the proposed approach.

model distillation [40], but it adopts convolutional neural network (CNN) as the student model, failing to inherit the knowledge learned in the teacher model parameters. The second challenge is *vocabulary compression*, because vocabulary indeed accounts for a substantial share of the model size, especially after the neural model is well compressed. Fig. 6.1 presents the notable proportion the vocabulary constitutes in BERT [17], a popular PLM. The proportion becomes much more substantial after the neural model is compressed by the recently proposed technique. Recent works addressing this issue are restricted only to [127] and [50], which propose techniques to obtain a small vocabulary during PLMs compression. However, both of them aim to train a task-agnostic model, with a minimum vocabulary size of 5000. As to be shown in our experiment, a task-specific model dedicated to intent detection has a smaller vocabulary.

To deal with the aforementioned two challenges, we propose a framework(Fig. 6.2). In specific, to push the limit of neural model compression with a few data, we augment CoFi [109], the SOTA transformers distillation method with novel utterances generated by off-the-shelf PLMs. Unlike [64], which adopts CNN as the student model, CoFi gradually prunes the teacher model parameters, keeping the learned knowledge as much as possible. Second, to reduce the vocabulary memory footprint, we design

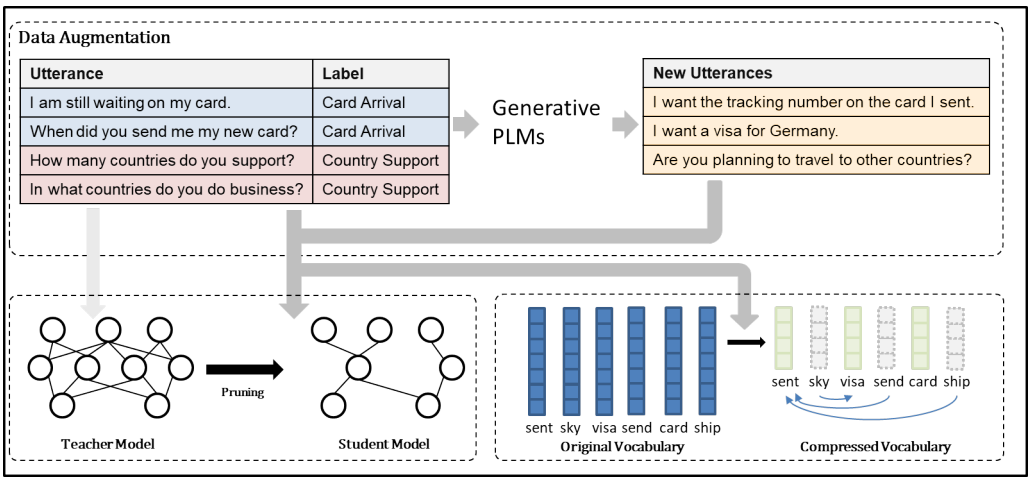


Figure 6.2: Illustration of the proposed model compression framework.

a novel vocabulary pruning method, V-Prune. Comprehensive evaluations are conducted to demonstrate the efficacy of the proposed methods. As shown in Fig. 6.1, under 5-shot scenario, our method reduces the neural model size by half compared against CoFi, and V-Prune reduces the memory footprint of the vocabulary by a factor of 30, while almost no loss in the performance is observed over various benchmarks.

6.2 Method

The target is to train a small intent classifier with a given small labeled set $\mathcal{D}$. To this end, we first train a large teacher model with $\mathcal{D}$ and then distill the learned knowledge into a small model.

6.2.1 Knowledge Distillation with Data Augmentation

Knowledge Distillation. To train the teacher model $\mathrm{f_t}$, we follow the common practice to attach a linear classifier on top of the $[CLS]$ representation of PLMs [17, 117] and optimize the parameters θ_t with $\mathcal{D}$ and cross-entropy loss function. Then, knowledge distillation is performed via aligning the logit output:

$$\theta = \arg\min_{\theta} \mathrm{KL}\left(\frac{\mathrm{f}\left(\mathcal{D};\theta\right)}{T}, \frac{\mathrm{f_t}\left(\mathcal{D};\theta_{\mathrm{t}}\right)}{T}\right), \tag{6.1}$$

where $\mathrm{KL}(\cdot)$ is the Kullback-Leibler (KL) divergence, $\mathrm{f}(\cdot)$ and $\mathrm{f_t}(\cdot)$ denote the output logit of the desired model and the teacher model, respectively. T is the tunable temperature parameter.

Vanilla knowledge distillation trains the student model from scratch, and hence results in sub-optimal performance (as shown in Table 6.1). In this work, we utilize CoFi [109], a recently proposed Transformer compression method. CoFi gradually prunes both coarse-grained modules and fine-grained parameters of BERT [17] to obtain the student model, and thus reach a promising performance. Like classic knowledge distillation, CoFi adopts the dataset $\mathcal{D}$ on which the output of the small model is aligned with the teacher model. However, once again, a significant performance drop is observed when $\mathcal{D}$ is small, as to be shown by the experiment.

Prompt:
The following sentences belong to the same category
'cancel transfer':
Example 1: How can i cancel a transfer I made?
Example 2: Cancel transaction.
Example 3: I need to cancel a transfer.
Example 4: I want to revert a transaction I did this morning.
Example 5: I made a mistake and performed a transaction
on the wrong account.
Example 6:

Generated Utterances:
Why has this transaction already been completed.
Is it possible to cancel an already invisible order?
I want to submit that transaction.

Figure 6.3: An example of the prompt and generated utterances under 5-shot scenario.

Data Augmentation. To alleviate the scarcity of the data for model compression, we propose augmenting $\mathcal{D}$ by off-the-shelf PLMs for two reasons. First, such PLMs do not need to be fine-tuned, which requires additional engineering effort and computation resources. Second, recently published off-the-shelf PLMs such as GPT-3 have shown promising results to generate texts with high quality. To prompt the PLMs to generate the desired utterances with the specific intent, we adopt the prompt in

Fig. 6.3, wherein some examples of the generated utterances are also given. Specifically, the prompt contains label name, such as "cancel transfer", followed by the utterances with the label. We sample 5 utterances with the label from $\mathcal{D}$ each time, compose the prompt and feed it into the PLM to generate one utterance each time. This process is repeated until enough utterances are collected. Finally, we filter out the utterances with unreasonable lengths.

6.2.2 Vocabulary Pruning (V-Prune)

It is intuitive that the original vocabulary with tens of thousand tokens is unnecessarily large for a given intent detection task. However, to obtain the task-specific vocabulary, two challenges persist. First, how to estimate the target vocabulary given tens of words in the few labeled utterances. Second, how to handle the missing tokens during inference. To tackle the first issue, we extract the most frequent K tokens in the *augmented dataset* generated in Section 6.2.1 to compose a vocabulary V'. It is a small fraction of the original vocabulary V. For the second challenge, we map the missing token to the nearest tokens in V':

$$\mathrm{M}(t) = \arg\min_{w \in V'} \mathrm{d}(t, w), \forall t \in V, \tag{6.2}$$

where $\mathrm{M}(t)$ denotes the map from any token t in V, to a token in V'. $\mathrm{d}(\cdot, \cdot)$ is the distance function, measuring the semantic distance between two tokens. We use Euclidean distance in our experiment. Fig. 6.3 gives an example of such mapping.

In addition, to further compress the vocabulary memory footprint, we adopt principal component analysis (PCA) transformation to reduce the dimension of word embeddings. The transformation is applied to map all word embeddings to the low-dimensional space, e.g. from 768 dimensions to 400 dimensions. During inference, the low-dimensional representation is mapped back to the original one by a simple linear mapping, before being fed into the model.

6.3 Experiments

6.3.1 Setup

Datasets. We adopt 4 large-scale practical benchmarks across various domains, including **BANKING77** [13], **HINT3** [3], **HWU64** [59] and **MCID** [2]. We randomly sample 5 data per label from the training partition to compose $\mathcal{D}$.

Our methods. We use the following models for data augmentation. **GPT-J-6B** [100] is an open-sourced generative auto-regressive text generation model with 6 billion parameters. **OPT-30B** [123] is a larger generative PLM with 30 billion parameters. **GPT-3-175B** [28] is one of the most powerful available generative PLMs with 175 billion parameters. **GPT-4-170T** [73] is the cutting-edge model with around 170 trillion parameters [51]. We adopt three architectures of student models. **CNN** [15] employs convolutional neural networks to extract the semantic feature of utterances, as in [64]. **BiLSTM** [34] uses the classic bidirectional long short-term memory networks. **CoFi** [109] is the recently proposed powerful transformer pruning method.

Details. We follow Section 3.3 to adopt OOS dataset for validation and hyper-parameters selection. We use $T = 10$ in Eq. 6.1. For V-Prune, we extract the top 2000 tokens and use the PCA transformation dimension 400. All experiments are performed with NVIDIA's A100 hardware and PyTorch framework.

6.3.2 Results

Data augmentation by generative PLMs is highly effective. As shown by Table 6.1 and Table 6.2, regardless of the student architecture, the data generated by PLMs play a key role in model compression under the few-shot scenario. The augmented data bring significantly better performance compared to baselines with only the few data. CoFi suffers a performance drop when the compression ratio goes

Method	BANKING77	HINT3	HWU64	MCID
BERT	68.69(1.39)	61.12(2.14)	74.72(1.40)	72.85(1.24)
CNN	56.44(.90)	51.21(1.55)	59.83(1.31)	55.40(2.04)
CNN + GPTJ-6B(ours)	63.52(1.20)	58.58(2.13)	70.26(.20)	69.40(.50)
CNN + OPT-30B(ours)	64.38(1.82)	58.04(.79)	69.41(1.21)	70.57(.59)
CNN + GPT3-175B(ours)	69.71(1.05)	57.58(1.29)	71.95(.56)	69.20(3.29)
CNN + GPT4-170T(ours)	63.37(1.69)	56.95(1.04)	69.68(1.93)	67.52(1.41)
BiLSTM	57.75(1.68)	50.98(1.54)	62.17(1.28)	60.53(2.90)
BiLSTM + GPTJ-6B(ours)	68.35(1.78)	58.43(1.52)	72.51(.85)	67.76(1.98)
BiLSTM + OPT-30B(ours)	69.21(2.07)	58.79(1.49)	71.53(.53)	67.68(2.43)
BiLSTM + GPT3-175B(ours)	68.60(2.20)	56.51(1.91)	72.05(.64)	65.95(3.84)
BiLSTM + GPT4-170T(ours)	67.39(1.64)	55.30(1.58)	70.54(1.20)	64.07(1.78)
CoFi	69.33(.02)	61.04(.02)	72.90(.07)	**73.96(.02)**
CoFi + GPTJ-6B(ours)	**70.67(1.90)**	**62.10(1.46)**	**74.59(1.55)**	**73.80(1.54)**
CoFi + OPT-30B(ours)	70.44(1.97)	**61.42(1.35)**	73.88(1.81)	72.36(2.12)
CoFi + GPT3-175B(ours)	**70.92(2.02)**	**61.04(1.41)**	**74.77(.96)**	**72.53(2.07)**
CoFi + GPT4-170T(ours)	**71.34(1.34)**	60.89(1.71)	**74.28(.53)**	72.07(2.66)

Table 6.1: Evaluation of data augmentation when compression ratio is 90%.

under 95%. However, with the augmented data, the loss can be almost eliminated. It is also found that the generative model size does not make a notable difference, although it is a popular belief that a larger model generates better texts – GPT-J with only 6 billion parameters plays on par with GPT-4, which is 30 thousand times larger. It is noteworthy that CoFi surpasses student models trained from scratch including CNN and BiLSTM, demonstrating the performance superiority of the technical choice over the work by [64].

V-Prune is effective. We apply V-Prune to CoFi+GPTJ-6B with the compression ratio of 95%. As shown in Table 6.3, a tiny fraction of the vocabulary is enough to keep a decent performance, following the intuition that task-wise vocabulary is small. Such a reduction in vocabulary size is necessary since it is the vocabulary

Method	BANKING77	HINT3	HWU64	MCID
BERT	68.69(1.39)	61.12(2.14)	74.72(1.40)	72.85(1.24)
CNN	55.90(1.30)	50.68(1.12)	59.69(1.14)	54.17(2.21)
CNN + GPTJ-6B(ours)	63.37(1.16)	59.62(2.45)	70.35(.60)	69.94(1.22)
CNN + OPT-30B(ours)	62.71(1.44)	58.14(.67)	69.24(.74)	69.61(1.15)
CNN + GPT3-175B(ours)	64.42(1.85)	57.69(1.15)	70.91(.87)	69.20(2.9)
CNN + GPT4-170T(ours)	63.79(1.58)	56.98(.66)	69.15(.52)	69.15(.52)
BiLSTM	59.07(1.28)	51.24(1.44)	61.94(1.87)	59.10(2.60)
BiLSTM + GPTJ-6B(ours)	68.48(1.86)	58.61(.66)	71.66(.82)	68.30(1.86)
BiLSTM + OPT-30B(ours)	68.81(2.14)	58.20(.93)	70.91(.62)	67.89(2.22)
BiLSTM + GPT3-175B(ours)	68.31(2.00)	56.78(1.46)	71.44(.57)	66.37(2.10)
BiLSTM + GPT4-170T(ours)	66.95(1.73)	55.74(1.31)	70.17(.73)	64.64(1.03)
CoFi	67.05(.02)	59.20(.03)	69.78(.01)	67.05(.02)
CoFi + GPTJ-6B(ours)	**70.38(1.80)**	**61.92(1.88)**	**73.78(1.14)**	**72.65(2.25)**
CoFi + OPT-30B(ours)	70.13(1.79)	**60.83(1.12)**	73.38(1.41)	71.29(1.37)
CoFi + GPT3-175B(ours)	**70.42(1.73)**	60.33(1.70)	**74.04(.70)**	**72.40(1.77)**
CoFi + GPT4-170T(ours)	**70.86(1.67)**	**60.74(1.32)**	**73.76(.63)**	**72.28(1.78)**

Table 6.2: Evaluation of data augmentation when compression ratio is 95%.

that occupies the most memory footprint after the model is well compressed, as shown in Fig. 6.1. In this experiment, V-Prune is configured to keep 3.4% vocabulary parameters. In the table, * denotes the vocabulary size of the original BERT, and † denotes CoFi+GPTJ-6B with a compression ratio of 95%. Additionally, we provide the ablation study result in Table 6.4, showing the efficacy of data augmentation and the nearest-neighbor replacement mechanism, respectively.

Impact of model size on performance. To obtain a deeper understanding, we visualize the impact of three hyper-parameters controlling the ultimate model size, including compression ratio (Fig. 6.4a), vocabulary token number (Fig. 6.4b) and word embedding dimension (6.4c). It is observed that even when the compression ratio is as high as 99%, we have a loss in the accuracy less than 3 percentage points.

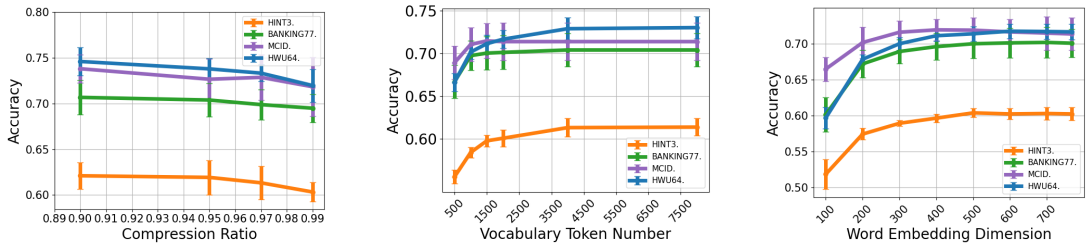
Size	BANKING77	HINT3	HWU64	MCID
100%*	68.69(1.39)	61.12(2.14)	74.72(1.40)	72.85(1.24)
99.7%†	70.38(1.80)	61.92(1.88)	73.78(1.14)	72.65(2.25)
3.4%	69.62(1.87)	59.65(.59)	71.13(.74)	71.95(1.50)

Table 6.3: Effectiveness of V-Prune.

5-shot	DA	NN	BANKING77	HINT3	HWU64	MCID
✓			67.79(1.80)	54.44(1.69)	67.03(.73)	61.15(2.99)
✓	✓		69.79(1.62)	58.85(1.17)	71.15(.62)	69.20(2.58)
✓	✓	✓	**70.12(1.96)**	**60.09(.97)**	**71.65(1.06)**	**71.38(2.19)**

Table 6.4: Ablation study of V-Prune. 5-shot denotes the small labeled dataset. DA denotes data augmentation using GPT-J. NN denotes the nearest-neighbor replacement mechanism.

As for the vocabulary size, when the token number decreases under 2000 and the dimension number under 400, the performance starts to drop drastically, confirming the conjecture that a small vocabulary is enough given the intent detection task. The result is significantly better than current works on vocabulary reduction, which yield a vocabulary size of 5000 [127, 50]. As a result, we obtain a well-performing intent classifier with around 5.1 million parameters, smaller than the original BERT by a factor of 21, making it convenient to deploy in resource-constrained scenarios.



(a) Model size. (b) Vocabulary size. (c) Embedding dimension.

Figure 6.4: The impact of hyper-parameters on the performance.

6.4 Conclusion

In this work, we identify the challenge of model compression for intent detection with a few data. We provide a simple but competitive baseline of the task, which combines the SOTA compression technique (CoFi) and data augmentation via generative PLMs. A novel vocabulary pruning technique is also proposed. The effectiveness of the method is demonstrated on four real-world benchmark datasets, showing that we can achieve a decent performance with only 5.1 million parameters, 21 times fewer than the original model.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis provides a comprehensive study of few-shot intent detection, regarding the aspect of transferability, expressiveness and efficiency.

Regarding the transferability, we demonstrate the feasibility of transferring across domains for few-shot intent detection. Specifically, when the unlabeled data is not available, we propose methods of transferring knowledge from source domains to the target domain. When the unlabeled data is available, we propose jointly utilizing the data in source domains and the target domain to train an intent classifier with competitive performance. We develop IntentBERT, a backbone network that is pre-trained with data from multiple intent detection domains. IntentBERT can significantly improve the performance in the target domain.

To boost the expressiveness of IntentBERT, we conduct a in-depth study of the anisotropy property of IntentBERT. It is found that supervised pre-training renders that feature space anisotropic, and isotropization hurts the performance after supervised pre-training. To mitigate the anisotropy, we design a framework of joint

training. Specifically, during supervised pre-training, we introduce two regularizers based on contrastive learning and correlation matrix, respectively, to regularize the feature space towards isotropy. Extensive experiments are conducted to demonstrate the efficacy of the proposed methods.

To further enhance the data-efficiency, we attempt to minimize the reliance on the extra data used in the continual pre-training stage. It is found that when fine-tuning with only the few data, the overfitting issue of PLMs may not be as severe as commonly believed. To better exploit the few data, we propose a framework comprising context augmentation and sequential self-distillation. Extensive experiments show the performance superiority of the proposed framework, given only two or more labeled samples per class. This framework is of superior data-efficiency because it does not exploit any extra data.

We finally focus on the computational efficiency of the solution. PLMs incur substantial computational overhead because of the substantial model size. we propose a model compression scheme that capitalizes on off-the-shelf generative PLMs for data augmentation. Moreover, we design a novel vocabulary pruning technique employing a nearest neighbour matching scheme. The proposed method manages to compresses the model by a factor of 21, and thus allows the deployment of the model in resource-constrained scenarios, such as mobile devices and embedded systems.

7.2 Future Work

7.2.1 Modular Task-oriented Dialogue Systems

A conventional task-oriented dialogue system consists of several independent modules, primarily encompassing intent detection, dialogue state tracking, slot filling and text generation. The future works on such systems include the following directions.

- Evaluation of industrial standard for all the aforementioned modules. Current evaluation datasets and protocols are mainly from the academic community, thereby disregarding several important properties of systems deployed in real-world scenarios. For example, the utterances may simultaneously have several intents [14], but most intent detection datasets assume single label for each utterance. To address the issue, it is imperative to provide industrial evaluation protocols, specifying details such as intent number, annotation cost, which encourages the research of more practical value.

- Efficient methods for dialogue systems. The introduction of PLMs have incurred significant computational overhead, thereby making it imperative to design dialogue systems consuming less computational power for resource-constrained scenarios. We have provided an early trial in Chapter 6 for intent detection. Efficient methods for other modules in a TOD system are to be explored.

- Multi-modal task-oriented dialogue system. Such a system encompasses not only textual data, but also images, speeches or videos. An example can be observed in a virtual reality system, where the TOD system consists of both utterances and videos [104]. The expansion of conventional TOD systems to multi-modal ones poses new challenges including spatial and temporal multi-modal reasoning, cross-modal co-reference, intent detection and slot filling considering multi-modal contexts.

7.2.2 New Era of Dialogue Systems

Most works of this thesis are undertaken prior to the era of large language models (LLMs). These models, such as GPT-4 [73], have been drastically changing the landscape of NLP. LLMs characterized by the vast scale and remarkable learning capabilities, have revolutionized the approach to building a dialogue system. Even without a dedicated intent detection module, LLMs have shown the promising capa-

bility of intent understanding and task fulfilling [74]. In this era, some of the future works are summarized as follows.

- Facilitating the modular design of TOD systems. Although LLMs make the end-to-end design plausible [74], the modular design presents distinctive advantages in terms of controllability, interpretability and compressibility. It is promising to harness the generative capability and the rich knowledge of LLMs to facilitate the construction of a modular system. Indeed, our effort in model compression in Chapter 6 is along the direction. Most recently, some LLMs have exhibited innate resilience to anisotropy [58], indicating the potential value in leveraging them to build modular TOD systems. However, exploiting these models to help design intent detection modules still faces challenges such as uncontrollable output and limited maximum input length.

- Efficient methods for model training and fine-tuning. Due to the cumbersome size, the training of LLMs consumes substantial computational resources, usually hundreds of enterprise-grade servers, thereby calling for techniques, including novel model architectures and training methods, to efficiently adapt LLMs to specific tasks such as intent detection.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[2] Abhinav Arora, Akshat Shrivastava, Mrinal Mohit, Lorena Sainz-Maza Lecanda, and Ahmed Aly. Cross-lingual transfer learning for intent detection of covid-19 utterances. 2020.

[3] Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. HINT3: Raising the bar for intent detection in the wild. In *EMNLP*, 2020.

[4] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *TACL*, 4:385–399, 2016.

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *NIPS*, 2000.

[6] Aditya Bhargava, Asli Celikyilmaz, Dilek Hakkani-Tür, and Ruhi Sarikaya. Easy contextual intent prediction and slot detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013.

[7] Hemanthage S Bhathiya and Uthayasanker Thayasivam. Meta learning for few-shot joint intent detection and slot-filling. In *ICMLT*, 2020.

[8] Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In *NAACL*, 2021.

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

[10] Hayet Brabra, Marcos Báez, Boualem Benatallah, Walid Gaaloul, Sara Bouguelia, and Shayan Zamanirad. Dialogue management in conversational systems: A review of approaches, challenges, and opportunities. *IEEE Transactions on Cognitive and Developmental Systems*, 14(3):783–798, 2022.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *NIPS*, 2020.

[12] Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *ICLR*, 2020.

[13] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*, 2020.

[14] Inigo Casanueva, Ivan Vulić, Georgios Spithourakis, and Paweł Budzianowski. NLU++: A multi-label, slot-rich, generalisable dataset for natural language understanding in task-oriented dialogue. In *Findings of NAACL*, 2022.

[15] Yahui Chen. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

[16] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[18] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv e-prints*, pages arXiv–2002, 2020.

[19] Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. ProtAugment: Intent detection meta-learning through unsupervised diverse paraphrasing. In *ACL-IJCNLP*, 2021.

[20] Thomas Dopierre, Christophe Gravier, Julien Subercaze, and Wilfried Logerais. Few-shot pseudo-labeling for intent detection. In *COLING*, 2020.

[21] John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. Gemini: A natural language system for spoken-language understanding. *arXiv preprint cmp-lg/9407007*, 1994.

[22] Jiangshu Du, Congying Xia, Wenpeng Yin, Tingting Liang, and Philip Yu. All labels together: Low-shot intent detection with an efficient label semantic encoding paradigm. In *AACL*, 2023.

[23] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *EMNLP*, 2018.

[24] Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[25] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *EMNLP-IJCNLP*, 2019.

[26] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[27] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[28] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

[29] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018.

[30] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *ICLR*, 2019.

[31] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

[32] Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. Induction networks for few-shot text classification. In *EMNLP-IJCNLP*, 2019.

[33] Allen L Gorin, Giuseppe Riccardi, and Jeremy H Wright. How may i help you? *Speech Communication*, 23(1-2):113–127, 1997.

[34] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[35] Jing Gu, Qingyang Wu, Chongruo Wu, Weiyan Shi, and Zhou Yu. PRAL: A tailored pre-training model for task-oriented dialog generation. In *ACL-IJCNLP*, 2021.

[36] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*, 2020.

[37] Patrick Haffner, Gokhan Tur, and Jerry H Wright. Optimizing svms for complex call classification. In *ICASSP*, 2003.

[38] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Visualizing and understanding the effectiveness of bert. *EMNLP*, 2019.

[39] Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of EMNLP*, 2020.

[40] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS 2014 Deep Learning Workshop*, 2015.

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

[42] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[43] Yutai Hou, Yongkui Lai, Yushan Wu, Wanxiang Che, and Ting Liu. Few-shot learning for multi-label intent detection. *AAAI*, 2021.

[44] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *WWW*, 2009.

[45] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of EMNLP*, 2021.

[46] Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems*, 38(3):1–32, 2020.

[47] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

[48] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[49] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[50] Alina Kolesnikova, Yuri Kuratov, Vasily Konovalov, and Mikhail Burtsev. Knowledge distillation of russian language models with reduction of vocabulary. *arXiv e-prints*, pages arXiv–2205, 2022.

[51] Anis Koubaa. Gpt-4 vs. gpt-3.5: A concise showdown. *Preprints*, March 2023.

[52] Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *EMNLP-IJCNLP*, 2019.

[53] Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *ICLR*, 2020.

[54] Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. Dialogue state tracking with a language model using schema-driven prompting. In *EMNLP*, 2021.

[55] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *EMNLP*, 2020.

[56] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. Exploiting bert for end-to-end aspect-based sentiment analysis. *EMNLP-IJCNLP*, 2019.

[57] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*, 2016.

[58] Bo Liu, Liming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. How good are large language models at out-of-distribution detection? *arXiv preprint arXiv:2308.10261*, 2023.

[59] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. In *IWSDS*, 2019.

[60] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[61] Tingting Ma, Qianhui Wu, Zhiwei Yu, Tiejun Zhao, and Chin-Yew Lin. On the effectiveness of sentence encoding for intent detection meta-learning. In *NAACL*, 2022.

[62] Alistair Martin, Jama Nateqi, Stefanie Gruarin, Nicolas Munsch, Isselmou Abdarahmane, Marc Zobel, and Bernhard Knapp. An artificial intelligence-based first-line defence against covid-19: digitally screening citizens for risks via a chatbot. *Scientific Reports*, 10(1):1–7, 2020.

[63] Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*, 2020.

[64] Luke Melas-Kyriazi, George Han, and Celine Liang. Generation-distillation for efficient natural language understanding in low-data settings. *EMNLP 2020 Workshop on Deep Learning for Low-resource NLP*, 2020.

[65] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In *NIPS*, 2022.

[66] Fei Mi, Yasheng Wang, and Yitong Li. Cins: Comprehensive instruction for few-shot learning in task-oriented dialog systems. In *AAAI*, 2022.

[67] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[68] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *NAACL*, 2013.

[69] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

[70] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *ICLR*, 2021.

[71] Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective post-processing for word representations. In *ICLR*, 2018.

[72] Hoang Nguyen, Chenwei Zhang, Congying Xia, and Philip Yu. Dynamic semantic matching and aggregation network for few-shot intent detection. In *Findings of EMNLP*, 2020.

[73] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

[74] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NIPS*, 2022.

[75] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[76] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. In *NIPS*, 2021.

[77] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020.

[78] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[79] Sara Rajaee and Mohammad Taher Pilehvar. A cluster-based approach for improving isotropy in contextual embedding space. In *ACL-IJCNLP*, 2021.

[80] Sara Rajaee and Mohammad Taher Pilehvar. How does fine-tuning affect the geometry of embedding space: A case study on isotropy. In *Findings of EMNLP*, 2021.

[81] Sara Rajaee and Mohammad Taher Pilehvar. An isotropy analysis in the multilingual bert embedding space. *Findings of ACL*, 2022.

[82] Suman Ravuri and Andreas Stolcke. Recurrent neural network and lstm models for lexical utterance classification. In *INTERSPEECH*, 2015.

[83] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, 2019.

[84] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[85] David E Rumelhart, James L McClelland, and CORPORATE PDP Research Group. *Parallel distributed processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT press, 1986.

[86] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[87] Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. In *ACL 2022 Workshop on NLP for Conversational AI*, 2022.

[88] Anna Sauer, Shima Asaadi, and Fabian Küch. Knowledge distillation meets few-shot learning: An approach for few-shot intent classification within and across domains. In *ACL 2022 Workshop on NLP for Conversational AI*, 2022.

[89] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[90] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(56):1929–1958, 2014.

[91] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

[92] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *CCL*, 2019.

[93] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

[94] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc., Series B (Methodological)*, 58(1):267–288, 1996.

[95] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *EAMT*, 2020.

[96] Gökhan Tür, Li Deng, Dilek Hakkani-Tür, and Xiaodong He. Towards deeper understanding: Deep convex networks for semantic utterance classification. In *ICASSP*, 2012.

[97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[98] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[99] Ivan Vulić, Pei-Hao Su, Samuel Coope, Daniela Gerz, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, and Tsung-Hsien Wen. ConvFiT: Conversational fine-tuning of pretrained language models. In *EMNLP*, 2021.

[100] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[101] Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *ACL*, 2022.

[102] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP*, 2019.

[103] Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *EMNLP*, 2020.

[104] Te-Lin Wu, Satwik Kottur, Andrea Madotto, Mahmoud Azab, Pedro Rodriguez, Babak Damavandi, Nanyun Peng, and Seungwhan Moon. Simmc-vr: A task-oriented multimodal dialog dataset with situated and immersive vr streams. In *ACL*, 2023.

[105] Congying Xia, Caiming Xiong, and Philip Yu. Pseudo siamese network for few-shot intent generation. In *ACM SIGIR*, 2021.

[106] Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. Composed variational natural language generation for few-shot intents. In *Findings of EMNLP*, 2020.

[107] Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *NAACL*, 2021.

[108] Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*, 2020.

[109] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. *arXiv preprint arXiv:2204.00408*, 2022.

[110] Canwen Xu and Julian McAuley. A survey on model compression for natural language processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[111] Puyang Xu and Ruhi Sarikaya. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU Workshop)*, pages 78–83, 2013.

[112] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *ACL-IJCNLP*, 2021.

[113] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NIPS*, 2019.

[114] Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *EMNLP*, 2021.

[115] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. In *NAACL*, 2018.

[116] Chenwei Zhang, Wei Fan, Nan Du, and Philip S. Yu. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *WWW*, 2016.

[117] Haode Zhang, Haowen Liang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Y.S. Lam. Revisit few-shot intent classification with PLMs: Direct fine-tuning vs. continual pre-training. In *Findings of ACL*, 2023.

[118] Haode Zhang, Haowen Liang, Yuwei Zhang, Li-Ming Zhan, Xiao-Ming Wu, Xiaolei Lu, and Albert Lam. Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization. In *NAACL*, 2022.

[119] Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. Effectiveness of pre-training for few-shot intent classification. In *Findings of EMNLP*, 2021.

[120] Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. Few-shot intent detection via contrastive pre-training and fine-tuning. In *EMNLP*, 2021.

[121] Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *EMNLP*, 2020.

[122] Li Zhang, Qing Lyu, and Chris Callison-Burch. Intent detection with WikiHow. In *AACL*, 2020.

[123] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[124] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. In *ICLR*, 2020.

[125] Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. New intent discovery with pre-training and contrastive learning. In *ACL*, 2022.

[126] Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17, 2020.

[127] Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. Extremely small BERT models from mixed-vocabulary training. In *EACL*, 2021.

[128] Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomáš Mikolov. Combining heterogeneous models for measuring relational similarity. In *NAACL*, 2013.

[129] Wenxuan Zhou, Bill Yuchen Lin, and Xiang Ren. Isobn: Fine-tuning bert with isotropic batch normalization. In *AAAI*, 2021.

[130] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.