A Framework for Effective Intermediate Task Selection in Transfer Learning

Alexander Pugantsov

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

December 2023

Contents

List of Tables

List of Figures

Abstract

This thesis investigates strategies to improve the performance of natural language processing (NLP) models across diverse tasks, particularly in environments with limited training data. Central to this investigation is the concept of transfer learning, a method where a model developed for one task is repurposed as the starting point for a model on another task. Determining which model will yield improved performance on a specific task is a complex and non-trivial challenge. This complexity arises due to the varying natures of tasks, the intricacies of model architectures, and the unpredictability of their interactions. Accurately estimating which models will be most effective before committing to extensive training can provide substantial benefits, including significant reductions in runtime, environmental impact, and other associated costs. To address this challenge, we propose a framework designed to determine, from a pool of candidate models, which one will provide the greatest performance enhancement for a given task. This framework consists of five components, each addressing a particular concern in selecting tasks for transfer. Parallel to this, and running continuously throughout the process, is the *Cost Estimation* background process. This module evaluates the resource efficiency of all other components, ensuring that the model development and adaptation processes are both effective and sustainable. The *Domain Adapter Generation* component involves developing resource-efficient models using training documents from various text-based tasks. The *Domain Transfer Analysis* component involves evaluating the models created in the previous stage on documents other than those they were originally trained on, providing an understanding in how these models perform on different types of textual data. The *Representation Construction* component involves the development of profiles or "representations" of each task based on, for example, terms or linguistic characteristics. These representations are intended to be expressive of the features of the underlying data, which we use in subsequent stages of our analysis. The *Divergence Estimation* component systematically quantifies the degree of variation between different representations through the use of statistical methods. By assessing the *divergence* between task-specific representations, this component helps identify which intermediate task models exhibit the most promising alignment for a specific target task. Finally, in the *Intermediate Task Selection* component uses the divergence data to rank tasks by their potential to improve model performance on a given target task. This

ranking provides guidance on which intermediate task models, when used to transfer to the target task, are most likely to yield the best performance.

In addressing the challenge of identifying tasks that are conducive to effective transfer learning, this thesis places a significant emphasis on evaluating representations against the performance scores derived from the Domain Adapter Generation stage. The core of this evaluation lies in assessing the "effectiveness" of these representations. Effectiveness, in this context, is defined as the capacity of representations to accurately estimate the most beneficial ordering of task combinations. This estimation is based on comparing the outputs of the divergence measures with the inherent ordering of tasks according to their *relative transfer gain*. Here, transfer gain is measured by the performance ranking of models that have been adapted from *intermediate* tasks to *target* tasks, where intermediate tasks are typically tasks abundant in training data, which are then used to transfer knowledge to resource-scarce target tasks that we would like to improve performance on. The theoretical basis of this approach is rooted in a fundamental principle of transfer learning: tasks with higher similarity in their representations are expected to offer greater improvements in model performance when transferred to a target task.

Consequently, the thesis investigates the relationship between task similarity, as quantified by our divergence measures, and the actual performance gains observed in transferred models. By analysing the correlation between divergence scores with the model performance rankings across tasks, we aim to validate the hypothesis that task similarity, in terms of representational divergence, is a key predictor of transfer success. This correlation not only provides a practical method to predict the projected effectiveness of task combinations but also offers insights into the nature of transfer learning itself, shedding light on which task characteristics most significantly impact model adaptability and performance enhancement. To evaluate the effectiveness of our framework, we simulate a scenario akin to a user employing the framework to "search" for the most suitable model to transfer to their specific target task. Traditionally, this process would involve exhaustive training and evaluation of all candidate tasks—often a time-consuming and resource-intensive process. We posit that, by predicting which tasks are likely to yield the largest performance gains ahead of time, through the analysis of task similarity, we can substantially improve the accuracy of task selection and also significantly reduce the time and resources required to find effective task pairs. Our framework allows users to bypass the labour-intensive cycle of trial and error, directly focusing on task combinations that are most likely to enhance their model's performance on a target task.

The central contributions of this thesis are the introduction of an effective and efficient intermediate task selection framework for transfer learning in natural language processing. This thesis draws from a diverse range of experiments, covering a broad range of NLP domains and experimental settings, to validate and refine the framework. The experiments

presented in this thesis demonstrate the potential of task selection approaches to provide more efficient, sustainable, and impactful practices in the field of transfer learning.

Acknowledgements

I would like to express my deepest gratitude to those who supported and encouraged me through the course of my PhD.

I would first like to thank my mother, who offered unwavering support through these difficult but exciting years, and my partner, Danielle, who helped me keep my priorities straight and more importantly, who reminded me that life existed outside of a code editor.

I would also like to honour the memory of my late father, grandfather, and grandmother. Though they are not here to share this accomplishment with me, their belief in me is not forgotten.

I am deeply grateful to my supervisor, Richard McCreadie, whose wisdom and guidance were only surpassed by his patience as I navigated my way out of dead-ends in my research.

Finally, I would also like to thank the TerrierTeam and Glasgow IR group, most of whom I only met after years of lockdown but got to know well after a few drinks. I will miss sharing a mutual panic over submission deadlines outside of the Sir Alwyn Williams Building.

Abbreviations

General

CO_2	Carbon dioxide emissions
kWh	Kilowatt hours
RT	Runtime

Probability and Information Theory

Cos.	Cosine
L_1	Manhattan distance
L_2	Euclidean distance
KL	Kullback-Leibler
JS	Jensen-Shannon
Bhat.	Bhattacharyya
Wass.	Wasserstein
CORAL	CORrelation ALignment
CMD	Central Moment Discrepancy
MMD	Maximum Mean Discrepancy

Machine Learning

NLP	Natural Language Processing
IR	Information Retrieval
LTR	Learning-to-Rank

NDCG	Normalised Discounted Cumulative Gain

Neural Networks

RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Network
ReLU	Rectified Linear Unit

Natural Language Processing

BOW	Bag-of-Words
CBOW	Continuous Bag-of-Words
NER	Named Entity Recognition
POS	Part-of-Speech
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency

Representations

TD	Term Distributions
LFD	Linguistic Feature Distributions
KFD	Topic (K-Means) Frequency Distributions
NE	Named Entity Distribution
DEP	Linguistic Dependency Distribution
UPOS	Part-of-Speech Distribution (Universal Tags)
XPOS	Part-of-Speech Distribution (eXtended Tags)
PWE	Probability-Weighted Embeddings
DCE	Distributive Contextual Embeddings
SEmb	Sentence-BERT Embeddings

Transfer Learning

I	Intermediate
T	Target
INST	Instance Aggregation
CTD	Centroid Aggregation
DOM	Domain Aggregation
TRM	Term Ranking Method
DC	Domain Context

Notation

Probability and Information Theory

P	Probability mass function
p	Probability density function
ρ	Spearman's rho
H	Entropy
μ	Mean
σ^2	Variance
$\tilde{\mu}_3$	Skewness
$\tilde{\mu}_4$	Kurtosis
D	Simpson's Index

Machine Learning

η	Learning rate
$\mathcal{T}$	Task
$\mathcal{D}$	Domain
θ	Parameters
$\mathcal{L}$	Loss function

Chapter 1

Introduction

Transfer learning has become increasingly prevalent in the development of models for Natural Language Processing (NLP). Drawing inspiration from observable biological learning mechanisms, this paradigm has become central to the development of modern neural language models. It is grounded in how animals, and indeed humans, repurpose acquired abilities to tackle new, related challenges, suggesting an inherent efficiency in transferring knowledge across different but related domains.

While traditional supervised learning approaches are effective, they often require large amounts of labelled data specific to each task which may not always be available. Transfer learning, by contrast, allows for leveraging pre-existing models trained on large datasets, mitigating the need for extensive data collection for each new task. This not only accelerates the development process but also addresses scenarios where obtaining task-specific data is challenging.

The heart of this approach lies in the concept of inductive bias—the preconceptions an algorithm holds about data distributions. Inductive transfer occurs when the knowledge gained from solving one problem is applied to a different but related problem. By applying models trained on sufficiently similar tasks, we introduce an inductive bias that influences the assumptions made by the model when faced with a different but related data distribution. This transfer of knowledge across tasks can improve the model's performance and generalisation capabilities on the target task. However, there remains an ever-present question that researchers in transfer learning have long strove to answer, *Why is pretraining useful for my task?* More specifically, *What information encoded in a pretrained model is transferable and advantageous for a given task?* Indeed, not all task combinations yield benefits such as performance improvements and the conditions underlying the effectiveness of transfer learning are not concretely understood.

Rosenstein et al. [88] highlighted that "the benefits of transfer learning depend, not surprisingly, on the similarity of the auxiliary and target tasks." However, determining task similarity is a non-trivial process. The subjective perception of similarity between

domains often does not correspond to their actual suitability for transfer, presenting a challenge in objectively quantifying task relatedness to effectively reflect their potential for successful transfer. Ben-David et al. [3] found that the performance on a target domain is largely bounded by its divergence to the source (or intermediate) domain. Domain divergence refers to the extent to which the characteristics of one domain, or task, differ from another. In the context of transfer learning, domain divergence can be used to identify tasks that are likely to be transferable. Tasks with lower divergence are typically more similar in nature and therefore more likely to share transferable features. Conversely, high divergence between tasks suggests less commonality and a lower probability of successful knowledge transfer.

In this thesis, we investigate how the process of transfer learning can be improved by identifying effective task combinations prior to training. The aim is to provide end-users with a ranking of the most effective intermediate tasks, designed to significantly reduce the time and resources spent on testing less effective task pairings. In particular, we focus on improving the ranking of intermediate tasks by identifying the characteristics that make certain tasks more transferable than others.

1.1 Motivation

As the accuracy of neural models continues to increase, so too does the computational cost of training and storing them. One approach of mitigating such cost is through using pretrained models to enhance performance on a downstream task, a paradigm commonly referred to as *transfer learning* [5,16,65]. Transfer learning has demonstrated its effectiveness in numerous applications such as representation learning [6], object recognition [27], and sentiment classification [112]. However, when and why transfer learning works is not concretely understood. Indeed, *it is not always clear what is signal and what is noise* [6] when transferring from one task to another. Traditionally, selecting the best tasks for transfer often involves an extensive trial-and-error process over many combinations and can quickly make the prospect of applying transfer learning undesirable. As such, it would be valuable to estimate whether a task pair combination will be effective pre-training, i.e. to estimate the *transferability* from one task to another.

Research in what constitutes transferable qualities between tasks indicate that domain divergence may play an important factor influencing transfer learning outcomes, that is, the degree to which domain pairs align or differ can significantly impact the transferability of knowledge between them [3]. Indeed, domain divergence has demonstrated promising results in different practical applications such as selecting document instances from one domain to improve performance in another [68, 91], learning domain-invariant representations [33], and predicting drops in performance when transferring from one task to

another [78, 106, 113].

The effectiveness of approximating divergence between domains is heavily influenced by how accurately the domains are represented. Representations of domains may be defined in relation to any set of events considered relevant to the task at hand. Traditional methods of representing text-based domains often include distributions of the relative frequencies of terms or linguistic dependencies in a corpus [68] or vector representations of words, known as word embeddings [4, 61], that allow for the encoding of rich semantic and syntactic information in a high-dimensional space. However, while these traditional methods have been foundational in representing text-based domains, there is limited exploration in establishing more complex methods of representation construction. In the estimation of transferability, further investigation into representation methods is particularly important, since representing domains in a manner suitable for evaluating transferability may require different properties compared to representations used for general language tasks. The need for a more comprehensive approach to transferability estimation, focused on creating and evaluating effective representations through domain divergence, leads us to identify the following three key challenges in this area:

Domain Representation. The ability to effectively estimate the similarity or divergence between representations hinges on how effectively they capture the characteristics of their underlying domains. Currently, representations are widely used [68, 91] in predicting the success of transfer, however, there is a lack of thorough investigation into why they work and how we can improve them. Most existing methods are applied without fully exploring their underlying mechanisms or potential for improvement. Hence, the first challenge we identify is: How do we construct domain representations that encapsulate domain characteristics more effectively? Furthermore, how do we evaluate the quality and accuracy of these representations in capturing relevant domain features?

Divergence Calculation. Accurately measuring the divergence between representations determines how well we are able to identify optimal tasks. Existing taxonomies provide a foundation [46], however, they lack a thorough investigation in applicability across diverse data types and scenarios. Hence, we identify the second challenge: How can divergence measures be used more effectively in diverse contexts? Additionally, how can we determine where and how these measures function best across different types of data and scenarios?

Intermediate Task Selection. Finding task combinations conducive to transfer learning often relies on intuition or trial-and-error. Transfer between dissimilar domains often leads to negative transfer [88] and the costs incurred searching through ineffective task combinations are expensive. Consequently, our third challenge is: How do we develop a systematic approach to accurately select intermediate tasks to optimise transfer-learning

efficiency? Specifically, how do we identify task combinations that minimise the risk of negative transfer and, thereby, reduce the costs associated with trial-and-error methods?

1.2 Thesis Statement

This thesis states that by constructing and evaluating effective domain representations using statistical divergence measures, we can significantly improve the accuracy of intermediate task selection in transfer learning. We propose a comprehensive framework focused on the systematic selection of intermediate tasks through the following functionalities: constructing and evaluating representations of domains, methods of calculating the divergence between representations, and strategies to rank and recommend optimal intermediate tasks for transfer. The core argument of this thesis is that a systematic, divergence-based approach to task selection will substantially reduce the time and resources required by exhaustive grid search methods. We hypothesise that by creating and comparing more effective representations of domains, we will be able to more accurately identify task pairs that are conducive to transfer. Finally, we posit that implementing these methodologies in a cohesive framework will lead to improved methodologies and practices in transfer learning.

1.3 Research Questions

RQ1: How can we effectively construct and evaluate domain representations to capture the characteristics and relationships between different domains, and to what extent can statistical divergence measures, applied to these representations, accurately estimate the transferability between tasks and improve the selection of optimal intermediate tasks for transfer learning?

This research question addresses the core of our thesis by focusing on the construction and evaluation of effective domain representations. We propose a comprehensive framework that encompasses various strategies for representation construction and evaluation, including distributional and embedding-based approaches. We evaluate the effectiveness of these representations by using statistical measures to estimate the divergence between representations and the performance scores from an extensive domain transfer analysis across 85 domains in different experimental settings. The expectation is that advanced domain representations and precise divergence estimations will significantly improve the accuracy of transferability estimation and effective intermediate task selection in transfer learning, as measured by the correlation between representation divergence and transfer performance metrics.

RQ2: How does a systematic, divergence-based approach to task selection compare to exhaustive grid search methods in terms of both accuracy and efficiency (time and computational resources), and how can we quantify and analyse the trade-offs between performance and efficiency to guide the development of more effective and sustainable transfer learning practices?

This question addresses the practical implications of our proposed framework by comparing it to both traditional exhaustive grid search methods and established baselines. We evaluate our systematic, divergence-based approach in terms of both accuracy and efficiency, using metrics such as NDCG, Regret, and rank-based analysis. To quantify the efficiency of our approach, we introduce a method for estimating the costs associated with model training and inference, focusing on runtime, CO_2 emissions, and energy consumption. By tracking resource consumption for each module, we provide insights into the computational efficiency and sustainability of transfer learning processes. The expectation is that our approach will substantially reduce the time and resources required by exhaustive grid search methods while maintaining high accuracy in task selection, as measured by the proportion of high-performing intermediate tasks found within the top predicted ranks and the associated cost savings in obtaining optimal transfer performance compared to exhaustive search.

1.4 Contributions

The main contributions of this thesis are the following. Firstly, we introduce a comprehensive framework for transferability estimation in natural language processing, designed to optimise the selection of intermediate tasks. This framework is composed of five modular components, each addressing a distinct aspect of the transfer learning process, along with an additional background process that monitors the efficiency and environmental impact of each step. The first two components involve the training and evaluation of intermediate and intermediate-to-target models, providing the foundation for the three components that directly tackle the challenges outlined in Section 1.1: developing and evaluating effective domain representations, quantifying task relationships through statistical divergence between these representations, and systematically predicting the most beneficial task ordering for a given target task. Our framework is therefore not only a pragmatic approach to task selection but also enhances the overall effectiveness and efficiency of transfer learning. We describe our contributions below:

Extensive Domain Transfer Analysis. We conduct a comprehensive analysis across 85 domains in transfer learning, encompassing three experimental settings based on the number of available target domain training samples: Zero-shot (0), Few-shot (50), and

Limited (1000). This extensive analysis, involving the evaluation of 6,700 individual task pairs, provides a broad and deep understanding of transfer learning effectiveness through the simulation of common, low-resource scenarios.

Strategies in Domain Representation Construction. Our work systematically evaluates both distributional and embedding representations, exploring a range of approaches from term distributions to frequency-weighted embeddings. Notably, we introduce a novel method for creating *Distributive Contextual Embeddings*, an approach that integrates the strengths of both distributional and embedding-based representations.

A Taxonomy of Divergence and Diversity Measures. We present a comprehensive taxonomy of divergence measures, offering a robust framework for both divergence calculation and the characterisation of individual distributions. Our taxonomy is designed to be adaptable, allowing for divergence calculations at various levels of representation abstraction. This flexibility enhances the utility of divergence measures, from broad comparisons at a higher level to more granular analyses at instance and centroid levels.

Systematic Transferability Estimation. Our framework introduces a method for estimating transferability through performance prediction. We thoroughly evaluate our approach against criteria of performance, efficiency, and the balance between these factors, providing a systematic and effective tool for selecting intermediate tasks in transfer learning.

Resource Tracking and Environmental Impact Assessment. We introduce a detailed method for estimating the costs associated with transferability estimation, focusing on both runtime and environmental impact. By tracking resource consumption for each module, our framework provides essential insights into the computational efficiency and sustainability of transfer learning processes.

Alongside our main contributions, we present an additional method employing an adapter-based approach for model training. This technique enhances storage efficiency and mitigates the risk of catastrophic forgetting in transfer learning. By training only select bottleneck layers, our method provides a more streamlined and efficient way of adapting models to new domains, ensuring the preservation of their foundational knowledge.

1.5 Recent Advancements in Transfer Learning

In recent years, the field of transfer learning has been significantly reshaped by the advent of *generative pre-training* [82], a methodology that fundamentally alters the traditional approaches used in both transfer learning and, more broadly, natural language processing.

Generative pre-training involves training a language model on a vast corpus of unlabelled text to learn broad representations of language patterns and structures. This approach forms the foundation for most modern large language models (LLMs) [13,19,82] and differs from targeted, supervised transfer learning—the focus of this thesis—in that, by learning from a wide range of unlabelled data, it allows the model to encode a generalised understanding of language, enabling them to perform various downstream tasks with minimal task-specific fine-tuning.

The generalised knowledge captured by LLMs has given rise to a new paradigm in transfer learning known as *In-Context Learning* (ICL) [13]. In traditional fine-tuning, a pre-trained model is adapted to a specific task by updating its weights using labelled data from that task. In contrast, ICL does not involve updating the model's weights; instead, it leverages the model's inherent knowledge by directly incorporating task-specific information into the input prompt. This is typically achieved by designing the prompt to include a few examples of the desired task, along with any necessary instructions or context. The model then uses this information to guide the decoder's generation process, effectively adapting its behaviour to "transfer" to the specific task without any explicit fine-tuning.

The emergence of ICL has far-reaching consequences for the field of transfer learning. By leveraging input prompts that provide examples of instructions and desired output, ICL circumvents the need for extensive task-specific fine-tuning, thereby reducing the associated costs. This is particularly valuable in scenarios where labelled data is scarce or where rapid adaptation to new tasks is required. However, it is important to recognise that supervision still plays an important in transfer learning, especially when tackling tasks of high complexity, as the necessary guidance may be too elaborate and multifaceted to be effectively conveyed within a concise input prompt. Under these circumstances, fine-tuning a pre-trained model using data that is sufficiently related to the target task can lead to superior performance.

Central to the success of both ICL and supervised transfer learning is the ability to accurately transfer the necessary instructions or knowledge from one task to another. This principle underscores an important challenge in both domains: our understanding of what constitutes relevant information often diverges from what a model deems useful or "transferable". This thesis investigates the factors that influence the effectiveness of knowledge transfer and, therefore, can inform the design of more effective in-context learning by providing insights into the specific task characteristics that lead to improved performance.

1.6 Thesis Outline

In this thesis, we explore how the construction and evaluation of effective domain representations, using statistical divergence measures, can enhance the accuracy of intermediate task selection in transfer learning. We introduce a comprehensive framework for estimating transferability, which focuses on creating domain representations, quantifying domain divergence, and predicting the most beneficial sequence of tasks for transfer learning. Using this framework, we investigate whether advanced domain representation and precise divergence estimation can lead to more accurate predictions of transferability. The outline of this thesis is as follows:

- In Chapter 2, we provide an overview of the relevant literature, required to understand the contents of this thesis. We review the fundamentals of information theory, machine learning, and neural networks. Furthermore, we discuss the development of language representations, provide a background on natural language processing, and define transfer learning in the context of our work, all of which assists in establishing the context of this thesis and supporting the motivation for our research.

- Chapter 3 presents our framework for transferability estimation. We describe each stage in the process: adapter generation and evaluation, our approach to representation construction, our divergence estimation approach, and finally, the models we train to rank intermediate tasks. We also include a section here on how we track resource consumption, from runtime to CO_2 emissions.

- In Chapter 4, we focus on the different divergence measures we use. This chapter is important for understanding the tools we use to estimate how similar different domains are for transfer learning. We categorise these measures into four groups: geometric, information-theoretic, statistical moments, and diversity measures. We explain how these measures work, why they are relevant for transfer learning, and their role in assessing our representations.

- Chapter 5 covers the experimental setup used for all successive experimental chapters. This chapter explains the basics needed for our study, starting with why we chose our datasets and how we transformed them for our use. It details our model training strategy, including adapter-based training, and how we optimised our process. We also discuss how we keep track of resource usage, from computing needs to CO_2 emissions. The chapter concludes with an analysis of how our models perform across various tasks, providing further motivation for our experiments.

- Chapter 6 presents our work in the creation of distributional representations. Here, we outline term-, category-, and topic-based distributions for each domain. Our

term-based distributions include the construction of term frequency (TF) and term frequency-inverse document frequency (TF-IDF) distributions and the various approaches to creating these effectively, from the construction of expressive vocabularies to the effectiveness of computing divergence at different levels of abstraction. Our category-based representations include creating frequency distributions of linguistic characteristics such as parts-of-speech (POS), named entities (NE) and linguistic dependencies (DEP). Using k-Means topic clustering, we also create topic frequency distributions (KFD) out of the resultant document-topic cluster assignments.

- Chapter 7 presents our work in the creation of embedding-based representations. Here, we compare and contrast the effectiveness and efficiency of computing static and contextual BERT [26] embeddings. We then incorporate information from our distributional representations by weighting terms and documents by term-based distributional probabilities to create Probability-Weighted Embeddings (PWE). Furthermore, we introduce a novel, vocabulary-informed approach of mapping contextual term embeddings to the vocabularies that form the basis of our term distributions, to incorporate both term significance and semantic similarity into a single representation, Distributive Contextual Embeddings (DCE).

- In Chapter 8, we introduce the main component of our framework, *Effective Intermediate Task Selection*. This chapter explains how we use distributional and embedding representations to select the best intermediate tasks for transfer learning. Our aim is to rank these tasks based on how well they transfer to a target task. We do this by training models with features derived from their divergences. The chapter outlines our methodology, explores key research questions, and evaluates the process by three criteria: Performance, Efficiency, and the balance between the two. In evaluating performance, we evaluate how often our model finds the best models quickly and complement this with an examination of how different representations impact model performance. In evaluating efficiency, we investigate the time and resources needed to build and evaluate these representations. Lastly, we compare the effectiveness of our model in finding optimal task combinations against the time taken to train all available models, comparing the trade-offs between performance and efficiency.

- In Chapter 9 we summarise our conclusions and contributions, discuss avenues for future work, and provide closing remarks.

1.7 Publications

The work presented in this thesis relates to the following peer-reviewed publications. The first is directly based on the research and results discussed throughout this thesis, while the second represents earlier work on interpretability and explainability methods that preceded and motivated the final direction taken:

1. **Pugantsov, A.** and McCreadie, R. (2022). Identifying Suitable Tasks for Inductive Transfer Through the Analysis of Feature Attributions. In *European Conference on Information Retrieval* (pp. 137-143).

2. **Pugantsov, A.** and McCreadie, R. (2023). Divergence-Based Domain Transferability for Zero-Shot Classification. In *Findings of the Association for Computational Linguistics: EACL 2023* (pp. 1649-1654).

In addition to the publications listed above, the following enumerated list represents TREC (Text REtrieval Conference) notebook papers that were published early on in the thesis work. These notebook papers document our participation in various TREC tracks and showcase the preliminary research and experiments that laid the foundation for the development of the main thesis contributions.

1. **Pugantsov, A.** and McCreadie, R. (2020). University of Glasgow Terrier Team (uogTr) at the TREC 2020 Incident Streams Track. In *TREC*.

2. **Pugantsov, A.** and McCreadie, R. (2021). University of Glasgow Terrier Team (uogTr) at the TREC 2021 Incident Streams Track. In *TREC*.

Chapter 2

Background

2.1 Introduction

The purpose of this chapter serves to provide readers with the necessary theoretical background to understand concepts used throughout this thesis. The chapter is structured into distinct sections, each focusing on a particular topic. The experiments in this work are based on the processing of textual data and rely heavily on the use of probabilistic methods and machine learning approaches. As such, we review the fundamentals of probability and information theory, discuss the foundations of machine learning and neural networks, the fundamentals of natural language processing which include how to represent language in a way that can be understood by machine learning algorithms and various tasks within the field. The chapter concludes with an overview of transfer learning, the primary domain of this research, outlining its significance and applications in natural language processing (NLP). The remainder of this chapter is structured as follows:

1. In Section 2.2, we provide an introduction to the mathematical fundamentals that underpin the methods we use to compare domains for transfer. This section covers the basics of probabilities and distributions, and the field of information theory.

2. Section 2.3 provides a discussion on the fundamentals of machine learning theory, covering the basics of how models learn from and generalise to input data, along with tasks and fundamental knowledge in machine learning that are used throughout this work: classification, regression, learning to rank, performance prediction, and cost estimation.

3. In Section 2.4, we discuss the basis of neural networks and their architectures, how neural networks facilitate machine learning through backpropagation, the concept of training models, and model optimisation methods.

4. Section 2.5 provides a background into natural language processing, discussing how

we represent textual data for processing with algorithms and common tasks in natural language processing that are relevant to this work.

5. Lastly, in Section 2.6 we provide a background on the field of transfer learning, including the motivations and common scenarios in which transfer learning is used, and provide an introduction to transferability estimation.

2.2 Probability and Information Theory

Probability measures the likelihood of an event occurring. In language-based processing, this translates to predicting word occurrence, sentence structures, or meaning from textual data. The following subsections will delve into the specifics of probability and information theory as they apply to NLP. We will explore the concept of a *Random Variable*, *Probability Mass/Density Functions*, and the concepts of *Joint*, *Marginal*, and *Conditional* Distributions. Additionally, we will discuss the principles of *Independence*, *Expectation*, and *Variance*. Integral to information theory, we will also examine *Entropy*, which quantifies the unpredictability or uncertainty in data, and *Information Gain*, reflecting the reduction in this uncertainty. Lastly, we briefly cover different categories of divergence measures, which we further expand on in our Taxonomy in Chapter 4. These elementary concepts in probability and information theory are foundational for understanding the statistical methods used in our work, particularly in the context of transfer learning.

2.2.1 Probability Basics

Random Variable. A *random variable* is a mathematical function that systematically assigns a real number to each outcome of a random process. Formally, given a probability space $(\Omega, \mathcal{F}, P))$, where Ω represents the set of all possible outcomes of an experiment, $\mathcal{F}$ is a collection of events that are subsets of Ω, and P is a probability measure assigning probabilities to these events. A random variable, denoted as X, is thus a function $X : \Omega \to \mathbb{R}$. It assigns a real number to each outcome in Ω.

Random variables can be either discrete or continuous. A discrete random variable takes on a finite set of values. In contrast, a continuous random variable may assume any value in an interval of numbers (e.g. body mass, height, or blood pressure). The importance of random variables lies in their ability to translate qualitative random processes into quantitative numerical values, enabling the application of mathematical and statistical methods to predict outcomes under uncertain conditions.

Probability Mass/Density Function (PMF/PDF). In probability theory, the behaviour of a random variable is characterised by its probability distribution, described

either by the *probability mass function* (PMF) for discrete variables or the *probability density function* (PDF) for continuous variables.

A probability mass function (PMF), often denoted as P, is used for discrete random variables, which take on a finite set of values. The PMF $P(x)$ maps a value x of the random variable X to the probability that X equals x. For any event x in the samples space of X, $P(x)$ adheres to the conditions $0 \leq P(x) \leq 1$, where an impossible event has a probability $P(x) = 0$ and a certain event has a probability $P(x) = 1$. The probabilities for all possible values of X sum up to to 1, i.e. $\sum_{x \in X} P(x) = 1$.

For continuous random variables, which can assume any value within a range, a probability density function (PDF) $p(x)$ is used. Unlike PMF, PDF does not give the probability of a random variable being exactly equal to a value. Instead, the probability of X lying within a small interval δx around x is approximated by $p(x)\delta x$. Since the exact probability at any point for a continuous variable is 0, the probability is considered over intervals. The integral of $p(x)$ over the entire range of X equals 1, ensuring normalisation: $\int p(x)dx = 1$.

Joint, Marginal, and Conditional Probability Distributions. Joint, marginal, and conditional probability distributions are interconnected concepts in probability theory, each serving a specific purpose in understanding and predicting outcomes based on multiple random variables.

A *joint probability distribution* is a measure that defines the probability of simultaneous occurrences of two or more random variables. A joint probability distribution, $P(x, y)$, quantifies the likelihood of events $x \in X$ and $y \in Y$ occurring simultaneously. Mathematically, for discrete random variables X and Y, the joint probability distribution is represented as $P(X = x, Y = y)$. In a machine learning context, consider a scenario where X is a feature indicating whether an email is flagged as important and Y is a feature indicating it contains a specific keyword. The joint distribution $P(X = x, Y = y)$ thus calculates the probability of both these features occurring together.

A *marginal probability distribution*, derived from a joint probability distribution, focuses on a single variable. For a random variable X, $P(x)$ is computed by either summing over all possible values of Y. For discrete variables,

$$P(X = x) = \sum_{y} P(x, y). \tag{2.1}$$

For continuous variables, we take the integral:

$$P(X = x) = \int_{y} p(x, y)dy. \tag{2.2}$$

In machine learning, understanding marginal probabilities can help in understanding the

distribution of individual features, independent of other variables.

A *conditional probability distribution* calculates the likelihood of an event given another event has occurred, that is, the probability of y given an event x, $P(y|x)$. We define this as the joint probability of events divided by the marginal probability of the event that has occurred:

$$P(y|x) = \frac{P(y,x)}{P(x)}. \tag{2.3}$$

The conditional probability distribution forms the foundations of many machine learning algorithms where predictions are based on known conditions.

Independence. *Independence* in probability theory is a fundamental concept describing the relationship between two or more random variables. Unlike conditional probability, where knowledge of one event influences the probability of another, independence implies a lack of such influence. Two random variables X and Y are independent if the occurrence of an event in X does not affect the probability of an event in Y, and vice versa. Mathematically, X and Y are independent if and only if:

$$P(X = x, Y = y) = P(X = x)P(Y = y) \tag{2.4}$$

for all x and y. This equation signifies that the joint probability of X and Y equals the product of their individual probabilities.

Expectation. Consider a random variable X with a PMF or PDF denoted as $P(X)$ and a function $f(x)$, we can determine the overall tendency of the function to take on certain values. This tendency or average value, often referred to as *expectation*, *expected value*, or the *mean* (μ) of the function $f(x)$ with respect to $P(X)$, quantifies the average behaviour of $f(X)$ when X follows the distribution $P(X)$. Mathematically, for discrete random variables, we can calculate this expectation using a summation:

$$\mathbb{E}_{x\sim P}[f(x)] = \sum_x P(x)f(x). \tag{2.5}$$

For a continuous random variable, we take the integral:

$$\mathbb{E}_{x\sim P}[f(x)] = \int p(x)f(x)dx \tag{2.6}$$

where x represents the possible values that X can take, and $P(x)$ or $p(x)$ is the probability associated with each value. The expectation essentially computes a weighted average,

where the weight is determined by the probabilities assigned to the distribution.

In machine learning, we often deal with random variables, such as the prediction errors of a model or the outcomes of random processes. Expectation helps us understand the typical behaviour of these variables. For instance, in linear regression, we aim to minimise the expected value of the squared difference between predicted and actual values, which corresponds to the Mean Squared Error (MSE).

Variance. *Variance* is a measure that quantifies the spread or dispersion of a random variable's values around its expected or average value. It provides insight into the variability or uncertainty associated with a random variable. In mathematical terms, the variance of a discrete random variable is often denoted $\text{Var}(X)$ or σ^2 and is calculated as:

$$\mu^2 = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]. \tag{2.7}$$

The mean and variance are also known as the *first* and *second moment*, respectively of a probability distribution, which we discuss further in Section 4.4.

2.2.2 Distributions

In probability theory, a probability distribution is a mathematical function or a rule that assigns probabilities to various outcomes or values that a random variable can take. In our work, we work exclusively with discrete probability distributions, using them to describe linguistic patterns in text. In this section, we will formally define and contextualise the types of discrete distributions used throughout.

A *Bernoulli distribution* [105] is a discrete probability distribution that models an experiment with a binary outcome: success (1) and failure (0). This distribution is characterised by a single parameter, often denoted as p, which represents the probability of success. The PMF of the Bernoulli distribution is, hence, defined as follows:

$$P(X = x) = p^x(1 - p)^{1-x}. \tag{2.8}$$

This distribution is often used in scenarios where there only exists two possible outcomes, such as success/failure. In this work, we conduct experiments using binary classification tasks, the output of which follows a Bernoulli distribution.

The *multinoulli distribution*, also known as the *categorical distribution*, is an extension of the Bernoulli distribution to cases where there are more than two possible outcomes. It is used to model discrete random variables within a finite number of distinct categories or classes. Mathematically, the PMF of a multinoulli random variable X is defined as:

$$P(X = x_i) = p_i \tag{2.9}$$

where $i \in \{i \mid 1 \leq i \leq n, i \in \mathbb{N}\}$, x_i represents the i-th category or class, p_i represents the probability of X being in category x_i, and n is the total number of categories. The multinoulli distribution, in machine learning, is often used in scenarios like classifying data into one of several discrete classes.

When dealing with a multinoulli or categorical distribution, it is common to normalise the distribution by its L_1 norm, which ensures that the probabilities are interpretable as relative frequencies, that is, that they sum to 1.

The L_1 normalisation of a multinoulli distribution involves dividing each probability by the sum of all probabilities. Mathematically, for a random variable X with probabilities $p_1, p_2, \ldots, p_n$, the L_1-normalised probabilities q_i are calculated as:

$$q_i = \frac{p_i}{\sum_{j=1}^{n} p_j} \tag{2.10}$$

where $i \in \{i \mid 1 \leq i \leq n, i \in \mathbb{N}\}$, ensuring that $\sum_{i=1} q_i = 1$. This normalisation approach is particularly useful when you want to interpret the probabilities as proportions of relative frequencies of different categories within the distribution, such as the occurrence of terms within a particular vocabulary.

2.2.3 Information Theory

Information Theory [96] is a mathematical framework for quantifying information, originally proposed for understanding and improving communication systems. Information Theory was designed to solve practical problems in the transmission of information over communication channels, aiming to determine the maximum amount of information that can be transmitted reliably and efficiently, addressing issues like signal degradation and noise in telecommunications.

Shannon entropy [96], $H(X)$ for a random variable X, is a measure of the uncertainty or randomness in the information context of X. It quantifies the expected "surprise" or uncertainty inherent in a random variable's possible outcomes. Higher entropy implies greater unpredictability, whereas lower entropy indicates more predictability. Mathematically, for a discrete random variable with numerous possible outcomes $x_1, x_2, \ldots, x_n$ and respective probabilities $P(x_1), P(x_2), \ldots, P(x_n)$, entropy is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i). \tag{2.11}$$

Entropy satisfies several key properties: it is non-negative, meaning $H(X) \geq 0$; it is maximal for a uniformly distributed random variable, where all outcomes are equally likely; and it is additive for independent random variables, i.e., $H(X, Y) = H(X) + H(Y)$ if X and Y are independent.

Information theory is fundamentally based on statistics and probability theory. The application of statistical and probability theory of information extends beyond individual distributions. In comparing the similarities and differences of distributional information, *divergence* measures, grounded in the same principles that inform information theory, allow us to quantify how one probability distribution diverges from another.

2.2.4 Divergence

In this section, we provide the foundational knowledge on the divergence categories used in this thesis. It is important to note that while we introduce these concepts, the full mathematical definitions and formulations of each measure are provided later in Chapter 4.

The concept of divergence is intrinsically linked to entropy and information theory. Kullback-Leibler [48] (KL) divergence, also known as *information gain*, quantifies the difference between two probability distributions in terms of their relative entropy. This concept extends not only to the measurement of uncertainty within a single distribution but also to the comparison of two different distributions. KL divergence represents a fundamental approach to quantify distributional differences as a member of the f-divergences [22] (and more broadly, *Information-theoretic* measures). However, its non-symmetric nature limits its utility as a distance metric, prompting the use of alternatives such as Jensen–Shannon divergence in this thesis.

f-divergences offer a generalised framework for comparing distributions using a convex function f. These differences have mathematical properties, such as non-negativity and consistency in different scenarios, which makes them particularly useful for machine learning applications. We also use α-divergences, which introduces a parameter α that adds flexibility to the divergence measure, allowing adjustments based on the specific application context. Rényi divergence [87], a variant of α-divergences, is used in our work because of its effectiveness in capturing subtle distributional differences.

In addition to the probabilistic measures of divergence, *Geometric* measures offer a spatial approach for understanding distributions. These measures focus on properties such as magnitude and direction by representing distributions in a high-dimensional space. This approach is particularly useful for comparing data such as embeddings, where the direction of vectors often encodes relevant semantic or syntactic information.

Complementing these approaches, *Optimal Transport* measures, such as Wasserstein-based [45] measures, provide a fundamentally different perspective on distribution comparison. It considers the cost or "work" needed to transform one distribution into another.

In addition to divergence measures, *Statistical Moments* offer another lens through which distributions can be evaluated. Moments capture the fundamental properties of distributions: the mean (first moment) describes their central tendency, whereas the variance (second moment) captures their spread. Higher moments, such as skewness (third moment) and kurtosis (fourth moment), provide insights into the distribution shape, revealing asymmetry and tail heaviness. In this work, statistical moments-based measures are used to compute divergence and to describe the characteristics of individual distributions.

Finally, our taxonomy also incorporates *Diversity Measures* to assess the variety and richness within individual distributions, complementing statistical moments measures. The inclusion of these measures in our taxonomy, inspired by their use in work by Ruder and Plank [91], include the aforementioned entropy-based measures and Simpson's Index [97], a measure typically used in ecological and biological studies to quantify the diversity of species.

2.2.5 Statistical Tests

Statistical tests are a fundamental aspect of data analysis in machine learning and various scientific disciplines. It is used to make inferences or draw conclusions about a population based on sample data. Statistical tests can be used to determine whether there is a significant difference between groups, whether a particular factor affects the outcome, or whether there is a correlation between variables. In the context of machine learning, statistical tests help in validating hypotheses about the significance of model improvement. Understanding the correct application of statistical tests ensures the reliability and validity of the conclusions drawn from the data.

Table 2.1: Interpretation of Spearman's ρ by Dancey and Reid [25], applied to both positive and negative relationships, i.e. focusing on the magnitude of the correlation.

Spearman ρ	Correlation
≥ 0.70	Very strong relationship
$0.40 - 0.69$	Strong relationship
$0.30 - 0.39$	Moderate relationship
$0.20 - 0.29$	Weak relationship
$0.01 - 0.19$	No or negligible relationship

One widely used statistical test in the analysis of rank data is Spearman's rank correlation coefficient [100], commonly known as Spearman's rho (ρ). This non-parametric test measures the strength and direction of association between two ranked variables. Spearman's rho is used when the data is ordinal or when the assumptions of the Pearson correlation coefficient (which requires a linear relationship) are not met. It assesses how well the relationship between two variables can be described using a monotonic function. To calculate

Spearman's rho, one must rank each set of data to be compared separately. Tied ranks are assigned the mean rank of the tied values. Spearman's rho is then computed according to the following formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.12}$$

where d_i is the difference between the ranks of corresponding values in the two datasets, and n is the number of observations. The coefficient ρ ranges from -1 to 1. A p value of $+1$ indicates a perfect positive correlation, while a p value of -1 indicates a perfect negative correlation. A value of 0 suggests no correlation. In our work, we establish a clear interpretation of Spearman's ρ, drawing upon guidelines set by Dancey and Reid [25] (Table 2.1). This interpretation focuses on the magnitude of the correlation, regardless of whether it is positive or negative. According to this interpretation framework, Spearman's ρ values equal to or greater than 0.70 are classified as indicating a "very strong relationship". In the context of our work, such values are considered highly indicative of effective relationships. Strong correlations, particularly in the aforementioned category, are deemed likely sufficient for application in production environments. Conversely, we consider magnitudes of less than 0.2 to indicate no correlation.

To determine the *significance* of an observed correlation, p-values are often used. P-values are essential in statistical hypothesis testing and calculate the likelihood of achieving results as extreme as, or more extreme than, what was observed under the assumption that the null hypothesis is true. The null hypothesis usually represents a standard position where there is no effect or difference. A small p-value (< 0.05) suggests that the evidence is strong enough to reject the null hypothesis. A p-value, however, does not confirm whether the hypothesis is true or false. Instead, it indicates the compatibility of observed data with null hypothesis.

In our work, we also use the Bonferroni [11] correction, an important method used to address the problem of multiple comparisons in statistical tests. When conducting multiple statistical tests, the likelihood of encountering a significant result due to chance (Type I error) increases. The Bonferroni correction is a conservative approach to control for this increased likelihood.

The Bonferroni correction adjusts the significance level (α) for each individual test in a family of comparisons. A "family" in this context refers to all comparisons that are being made in a particular analysis. The adjusted significance level is calculated by dividing the original α value by the number of tests being performed. For instance, if α is set to 0.05 for a family of 10 comparisons, the Bonferroni corrected α for each individual test becomes $\frac{0.05}{10} = 0.005$. A test result is only considered statistically significant if its p-value is less than the adjusted α threshold.

While the Bonferroni correction is useful for reducing the risk of Type I errors, it is known for being very stringent. This stringency can sometimes lead to an increase in Type II errors (failing to reject a false null hypothesis). Therefore, the use of Bonferroni correction is carefully considered in our work, balancing the need to avoid false positives while maintaining the ability to detect true effects.

2.3 Machine Learning

To fully engage with the material presented, it is essential to grasp some fundamental machine learning concepts, as they recur throughout this thesis. Machine learning, a field at the intersection of computing science and statistics, focuses on creating and using mathematical models to interpret data and make decisions. The core of machine learning involves constructing models, mathematical representations of the processes we aim to understand or predict, based on input data.

In this thesis, we primarily explore two branches of machine learning: *supervised* and *unsupervised* learning. Supervised learning occurs with labelled data, where each data point in the training set is paired with a corresponding label or output. Mathematically, for each input $x_i \in \mathbb{R}^d$ in our dataset, there is an associated label y_i, leading to pairs inputs (x_i, y_i) in the training data. Here, x_i denotes the input data point, d denotes the dimension of the input space, and y_i represents the expected output or label. The aim in supervised learning is then to develop a function f that maps inputs to outputs as $f : X \to Y$, effectively training the model to predict or categorise data.

Unsupervised learning, in contrast to supervised learning, operates on unlabelled data. In this approach, the dataset consists of inputs $x_i \in \mathbb{R}^d$ without corresponding output labels. The goal in unsupervised learning is to discover underlying structures or patterns within the data. Mathematically, this involves identifying functions or mappings that can capture these hidden structures. For example, a function $g : X \to Z$ may be used, where Z represents a new representation or structured discovered within the data X. This process does not rely on predefined labels but instead seeks to uncover relationships, clusters, or other meaningful insights directly from the input data X.

In the context of this thesis, we examine how both supervised and unsupervised learning methodologies can be effectively applied in natural language processing. Our focus in supervised learning encompasses tasks such as *classification*, where data points are assigned to categories; *regression*, which involves predicting continuous values; and *learning-to-rank*, focusing on ordering items based on relevance. In the context of unsupervised learning, we will explore clustering, a method of grouping data points based on similarity or other criteria, which plays a significant role in understanding and interpreting unstructured data.

2.3.1 Classification

Classification in supervised learning is the process of assigning data points to predefined categories. This concept is essential for understanding many machine learning applications, as it deals with how models discern and categorise information. Two primary forms of classification are binary and multi-class classification.

Binary classification involves categorising each instance in the dataset into one of two distinct classes. This task can be represented as learning a function $f : \mathbb{R}^d \to \{0, 1\}$. For a given input vector $x_i \in \mathbb{R}^d$, the model predicts a binary label y_i, either 0 or 1. In multi-class classification, the objective is to assign each input to one of N classes. This is represented as a function $f : \mathbb{R}^d \to \{1, 2, \ldots, N\}$. For an input vector x_i, the model outputs a set of scores for each class.

The prediction mechanism typically involves a linear combination of the input features weighted by a parameter vector $\theta \in \mathbb{R}^d$, plus a bias term $b \in \mathbb{R}$:

$$\hat{y}_i = \theta^T x_i + b \tag{2.13}$$

where $\hat{y}_i$ is the predicted value of example i. In binary classification, the linear output $\hat{y}_i$ of the model is often passed through a sigmoid function, denoted as $\sigma(z)$. The sigmoid function maps any real-valued number into a value between 0 and 1, effectively transforming the output into probabilities. The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{2.14}$$

Applied to our model's output, $\sigma(\hat{y}_i)$, this transformation converts the linear score into a probability of the input x_i belonging to one of two classes. The output after applying the sigmoid function represents a Bernoulli distribution, as it models the probability of a binary outcome. For multi-class classification, the generalisation of the sigmoid function, known as the *softmax* [10,12] function, is used. The softmax functions converts a vector of values into a probability distribution over predicted output classes. For a vector of scores Z (corresponding to the scores for each class), the softmax function for the j-th class is given by:

$$\text{softmax}(Z)_j = \frac{e^{Z_j}}{\sum_{k=1}^{N} e^{Z_k}} \tag{2.15}$$

where N is the total number of classes. This function ensures that the sum of the probabilities of all output classes is 1, making the outputs interpretable as a probability distribu-

tion. The result of applying the softmax function to the model's outputs is a multinoulli (or categorical) distribution, suitable for multi-class scenarios where each input can be categorised into one of multiple classes.

Evaluating classifiers involves assessing the accuracy of its predictions, providing insights into how well the model is likely to perform on unseen data, guiding improvements and adjustments. The evaluation of classifiers is primarily focused on how accurately these probabilistic predictions align with the actual labels of the data. Predictions are categorised into four groups: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). True positives and true negatives represent the instances where the model correctly predicts the positive and negative classes, respectively. Conversely, false positives are instances incorrectly labelled as positive, and false negatives are positive instances that the model incorrectly labels as negative.

In the context of binary classification, used throughout this work, a common approach to evaluation involves using metrics such as *precision*, *recall*, and *F-score*. These metrics provide an understanding of a model's performance in terms of both its accuracy and reliability in predicting positive instances.

Precision is a metric that measures the model's accuracy in classifying an instance as positive. Formally, it represents the ratio of correctly predicted positive observations to the total predicted positive observations. The precision formula is given by:

$$Precision = \frac{TP}{TP + FP} \tag{2.16}$$

In this equation, True Positives (TP) are the correctly identified positive cases, and False Positives (FP) are the cases where the model incorrectly predicted positive outcomes. High precision indicates that the model has a low rate of false positive predictions. This metric is particularly important in scenarios where the cost of falsely identifying negatives as positives is high.

Recall measures the model's ability to correctly identify positive instances. This value is defined as the ratio of correctly predicted positive observations to all observations in the actual class. The recall formula is:

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

Here, False Negatives (FN) represent the instances that are actually positive but were incorrectly predicted as negative by the model. A high recall score indicates that the model is effective in identifying positive cases and has a low rate of false negatives. This metric is particularly important in scenarios where missing a positive instance can have severe consequences.

Precision and Recall are often used together to provide a more comprehensive assess-

ment of a model's performance. While precision focuses on the correctness of positive predictions, recall addresses the model's ability to identify all relevant instances.

The F-score, a general term for the F-measure, is a set of metrics that blend precision and recall, two critical measures of a model's accuracy, into a single score. The general formula for the F-score is given by:

$$F - score = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall} \tag{2.18}$$

In this formula, β is a parameter that determines the relative weight of precision and recall. A higher β values the recall more, while a lower β emphasises precision. This flexibility allows for adjustments based on the specific requirements of different applications. The F_1-score is a specific instance of the F-score where β is set to 1, giving equal weight to precision and recall. It is calculated as the harmonic mean of precision and recall:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.19}$$

By combining precision and recall into a single metric, the F_1-score provides a succinct and effective measure of a model's accuracy, especially in cases where the cost of false positives and false negatives is roughly equivalent.

2.3.2 Regression

Regression tasks in supervised learning are distinct from classification in that they aim to predict continuous or quantitative outcomes. Where classification categorises data into discrete classes, regression models the relationship between input variables and a continuous output variable.

Linear regression, the simplest and most widely used form of regression, assumes a linear relationship between the input variables and the output. It uses a similar prediction mechanism to that in linear classification (as referenced in Equation 2.3.1), but with a focus on predicting continuous values rather than categories. For a given input vector $x_i \in \mathbb{R}^d$, linear regression predicts a continuous output value $\hat{y}_i \in \mathbb{R}$ using the same linear combination of input features.

In terms of evaluation, while measures like precision and recall are central to classification tasks, regression models are often evaluated using different metrics. Common metrics include *Mean Squared Error* [34] (MSE) and *Root Mean Squared Error* (RMSE), which quantify the difference between the predicted and actual values. However, in this work, linear regression is not evaluated directly through these metrics but is instead used as a proxy for *learning-to-rank* tasks, a methodological choice reflecting the specific require-

ments and context of our analysis.

2.3.3 Learning to Rank

Learning-to-Rank [53] (LTR) is a type of machine learning problem used in situations where the goal is to automatically sort items into a meaningful order. It is particularly prevalent in applications like search engines, where the objective is to rank web pages based on their relevance to a user's query, and recommendation systems, where the aim is to present items, such as products, in an order that reflects the user's preferences.

In Learning-to-Rank, the input typically consists of a set of items and features associated with each item-query pair or item-user pair. The task is to learn a ranking function $f : X \to \mathbb{R}$. For a given input vector $x_i \in \mathbb{R}^d$, representing the features of an item, the model assigns a score indicating the item's relevance or importance.

There are three main approaches to Learning-to-Rank: *pointwise*, *pairwise*, and *listwise* approaches. The pointwise approach treats ranking as a regression or classification problem, predicting a numerical score or a categorical relevance level for each item independently. The pairwise approach, on the other hand, focuses on correctly ordering pairs of items, effectively transforming the ranking problem into a binary classification problem where the model learns to predict which item in a pair is more relevant. Finally, the listwise approach considers the entire list of items, aiming to optimise the order of the entire list directly.

Although LTR is fundamentally different from classification and regression, it shares the use of a linear combination of features weighted by a parameter vector, as outlined in Equation 2.3.1. The model's output, however, is not a discrete class label or a continuous value but a relevance score used to rank items. In terms of evaluation, LTR models are assessed differently compared to classification or regression models. The evaluation focuses on the quality of the ranking produced. In this thesis, we primarily assess the quality of rankings using *Normalized Discounted Cumulative Gain* [43] (NDCG).

NDCG is based on two concepts: *Cumulative Gain* (CG) and *Discounted Cumulative Gain* (DCG). CG is the sum of the graded relevance values of all items in a list up to a particular rank position. DCG introduces a discounting factor to CG, giving higher relevance to items at the top of the list and less to those further down.

Given a list of items, each with a relevance score, the DCG at a particular rank position k is calculated as:

$$\mathrm{DCG}_k = \sum_{i=1}^{k} \frac{2^{\mathrm{rel}_i} - 1}{log_2(i+1)} \tag{2.20}$$

where rel_i is the relevance score of the item at position i, and the denominator $log_2(i+1)$

is the discounting factor, which logarithmically reduces the contribution of items as their rank position increases. To normalise the DCG, allowing for comparison across different sets of results, it is compared to the *Ideal Discounted Cumulative Gain* (IDCG), which is the DCG value obtained from the ideal ranking of items. The NDCG at rank k is defined as:

$$\text{NDCG}_k = \frac{\text{DCG}_k}{\text{IDCG}_k} \tag{2.21}$$

The choice of k in NDCG is important as it sets a cut-off rank in the list, beyond which items are not considered in the evaluation. This cut-off is significant because users typically focus on the top results. By varying k, one can analyse the effectiveness of the ranking algorithm at retrieving relevant items within the top k results. For instance, NDCG@10 evaluates the quality of the top 10 items in the ranking, aligning with common user behaviour in scenarios like web searches.

2.3.4 Clustering

Clustering is a type of unsupervised learning that focuses on grouping a set of objects in such a way that objects in the same group, known as a cluster, are more similar to each other than to objects in other groups. *K-means* [56, 101] clustering categorises n observations into k clusters, where each observation belongs to whichever cluster has the nearest mean. The algorithm involves the following steps: (1) k initial centroids are chosen, either randomly or through some heuristics; (2) each data point is assigned to the nearest centroid, creating k clusters; (3) the centroids are recalculated as the mean of all data points in the cluster; (4) the assignment is then repeated and updated until the centroids no longer change substantially.

Mathematically, given a set of observations $x_1, x_2, \ldots, x_n$, k-means clustering partitions each observation into $k(k \leq n)$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimise the sum of squared distances between each point in a cluster and the centroid of that cluster. In other words, its objective is to find:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{2.22}$$

where μ_i is the mean of points in S_i. Evaluating the performance of k-means clustering is challenging, particularly since the ground truth labels are not known. One of the methods to evaluate clustering in such scenarios is the *silhouette analysis* [89]. In the context of clustering algorithms, the silhouette score measures how similar a particular data point is to its own cluster compared to other clusters, in effect, measuring the quality or separation

of of clusters. The silhouette score for a single data point is calculated as follows:

$$\text{Silhouette Score} = \frac{b - a}{\max(a, b)} \tag{2.23}$$

where a and b denote the average distance of that data point to other points within the same cluster and the smallest average distance from that data point to points in other clusters, respectively. Silhouette scores range from -1 to 1, where values closer to 1 mean that the data point is a good match for its own cluster and separable from neighbouring clusters.

2.3.5 Performance Prediction

Performance Prediction refers to estimating the effectiveness of a given model before it is applied. This task spans multiple fields, including information retrieval, recommendation systems, and natural language processing.

In information retrieval, one of the key challenges has been predicting the performance of queries, where effective query prediction methods can improve retrieval accuracy. Cronen-Townsend et al. [21] approached the problem by focusing on a query's *clarity*. Their metric, termed a *clarity score*, is derived using the Kullback-Leibler divergence, comparing the language model of a specific query against the broader model of the entire document collection. The underlying principle of their method is that queries with a higher degree of specificity or clarity score, are more likely to contribute to better retrieval performance. Similarly, He and Ounis [39] adopt a computationally efficient approach by using query performance predictors, such as query length, clarity, and scope, that can be computed before the retrieval process.

Bellogín and Castells [8] adapt these clarity-based predictors. They introduced *neighbor goodness*, a measure reflecting the impact of including or excluding a user's ratings on the system's performance. Paun et al. [76], noting the enormous costs incurred in training recommender systems, predict their efficiency in terms of training time and memory costs.

Central to these approaches is the quantification of some similarity or divergence to predict performance. Indeed, in the context of transfer learning, the *transferability* between domains is often defined as a function of their domain divergence. Much like query performance prediction informs retrieval strategies, predicting the success or failure of transfer, in terms of their task performance, informs effective transfer learning strategies. Van Asch and Daelemans [106] approximate the distance between domains to predict performance degradation across domains. Similarly, prior work [29, 78] leverages domain similarity to predict performance drops in cross-domain sentiment classification. Xia et al. [113] use statistical measures of divergence in machine translation to predict the performance of models on new languages. In our work, we approach performance prediction as a ranking

task with the goal of predicting the relative performance levels of different models.

2.3.6 Cost Estimation

The rapid growth and widespread adoption of machine learning and deep neural networks have led to significant advancements across various domains. However, the training of large-scale machine learning models comes with a substantial environmental cost. In this section, we explore the environmental impact of training machine learning models and discuss the need for more sustainable practices. We then introduce several tools and approaches that have been developed to monitor resource consumption in model training and inference.

Machine learning models, particularly deep learning architectures, require vast amounts of computational resources and energy during the training process. The carbon footprint associated with training these models has become a growing concern in the scientific community. Strubell et al. [102] estimate that the cost of training BERT_{base}—originally trained on 8x NVIDIA P100 GPUs [108]—can emit almost as much carbon dioxide emissions as a flight from New York to San Francisco. Furthermore, they report that the neural architecture search for machine translation and language modelling conducted by So et al. [98], which required 274,120 GPU hours on 8x NVIDIA P100 GPUs, emits nearly as much carbon dioxide as five cars (including fuel) over their lifetime. These alarming statistics highlight the urgent need to consider the environmental impact of machine learning practices and develop more sustainable approaches.

To address this challenge, various tools have been developed to measure the energy footprint of computing [1, 55, 71, 102]. In this work, we make use of the CodeCarbon [55] python package to measure the energy expenditure of our experiments. In this work, we employ CodeCarbon [55], a Python package that estimates the carbon dioxide equivalent (CO_2eq) emissions generated by the training or inference process. CodeCarbon captures relevant metrics such as power consumption, duration, and the geographical location of the computing resources, combining this information with the carbon intensity of the energy grid to provide an estimate of the environmental impact.

We use CodeCarbon to assess the energy consumption and carbon footprint of our proposed methods for transferability estimation in natural language processing. By tracking the environmental impact of constructing different domain representations (e.g., Term Distributions, BERT embeddings) and training transfer learning models across various datasets and settings, we aim to provide insights into the computational efficiency and sustainability of our approach. This allows us to make informed decisions about the use of different representations in our framework and compare the environmental cost of our method to traditional approaches like exhaustive grid search.

The insights gained from cost estimation contribute to the broader goal of developing

sustainable practices in machine learning and natural language processing. By quantifying the environmental impact of our methods and raising awareness of the carbon footprint of different techniques, we aim to encourage the development of energy-efficient and environmentally friendly approaches to transfer learning and task selection.

2.4 Neural Networks

In this section, we transition into a discussion on neural networks and their relation to our work. Understanding the fundamentals of neural networks is important for for the reader as they underpin the models and techniques discussed throughout. Neural networks differ from simpler models by their ability to learn intricate patterns in data. Unlike a linear model that uses a straightforward combination of inputs, neural networks consist of layers of interconnected nodes, or *neurons*, each performing a simple computation. This layered architecture enables neural networks to solve tasks that are too complex for simpler models.

Recall from Equation 2.3.1 the prediction mechanism of linear models: $\hat{y}_i = \theta^T x_i + b$, where θ represents a weight vector, x_i is the input vector, and b is the bias term. The output $\hat{y}_i$ is a linear combination of the inputs. Neural networks extend this by introducing multiple such combinations in one or more hidden layers ($\mathbf{h}$). Each layer contains a number of neurons, and each neuron performs a computation similar to the linear equation but in a more complex network structure. For a single hidden layer, the computation can be represented as:

$$\mathbf{h} = g(\mathbf{W}^T x_i + \mathbf{b}) \tag{2.24}$$

where $\mathbf{W}$ is the weight matrix corresponding to the connections between the input layer and the hidden layer, and $\mathbf{b}$ is the bias vector for the hidden layer, and g is an *activation function*. The activation function, applied element-wise to the vector, introduces non-linearity into the model. This non-linearity is important for the network's ability to learn complex relationship. Common activation functions include the aforementioned sigmoid (σ) and the Rectified Linear Unit [32] (ReLU):

$$\sigma(x) = max(0, x). \tag{2.25}$$

In deeper networks with multiple hidden layers, the output of one layer feeds into the next as input, continuing until the output of the entire network is produced, known as *forward propagation*.

2.4.1 Models, Layers, and Weights

Neural network models are essentially a series of mathematical operations defined by their architecture. Each model consists of an input layer, multiple hidden layers, and an output layer. The input layer receives the data, while the output layer produces the final result—typically transformed via a sigmoid or softmax function to produce a Bernoulli or categorical distribution, respectively. Hidden layers, situated between the input and output layers, are where most of the model's computation occurs. This section will discuss the models used throughout this work and their architectures.

Recurrent Neural Networks. Building on the foundational idea of neural networks as a series of mathematical operations, Recurrent Neural Networks [92] (RNNs) introduce the capability to process sequences of inputs. The defining feature of RNNs is their internal memory, which captures information about previous inputs. In an RNN, the core component is the hidden state, $\mathbf{h}_t$, which acts as the network's memory at each time step t. This hidden state captures information about the sequence processed up to that point, effectively allowing the RNN to "remember" previous inputs in the sequence. At each time step, the RNN updates its hidden state using the current input and the previous hidden state:

$$\begin{aligned} \mathbf{h}_t &= \sigma_h(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h\mathbf{h}_{t-1} + \mathbf{b}_h) \\ \mathbf{y}_t &= \sigma_y(\mathbf{W}_y\mathbf{h}_t + \mathbf{b}_y) \end{aligned} \tag{2.26}$$

where $\mathbf{x}_t$ represents the input at time step t, $\mathbf{W}_h$ and $\mathbf{U}_h$ are weight matrices applied to the current input and the previous hidden state, respectively, and $\mathbf{b}_h$ is a bias term. The function σ_h is an activation function applied to the linear combination of these terms to produce the new hidden state $\mathbf{h}_t$. The second equation describes how the RNN generates the output $\mathbf{y}_t$ at each time step. Here, $\mathbf{W}_y$ is the weight matrix, and $\mathbf{b}_y$ is the bias term for the output layer.

This mechanism allows the RNN to pass information across time steps, making it suitable for tasks involving sequential data like language processing. However, standard RNNs often struggle with learning long spans of text. This limitation led to the development of more advanced RNN architectures like *Long Short-Term Memory* networks (LSTM), which incorporate mechanisms to better capture these long-range dependencies.

Long Short-Term Memory Networks. Long Short-Term Memory Networks [41] (LSTMs) are more adept at capturing longer sequences of text by introducing contextual state cells which preserve the historical context of the input sequence. Components known as *gates* decide how much information is to be forgotten from earlier stages in the sequence (*forget gate*), how much of the new information flowing into each cell is to be

retained (*input gate*), and what information is important enough to output to the next state (*output gate*):

$$\begin{aligned}
\mathbf{f}t &= \sigma(\mathbf{W}f \cdot [\mathbf{h}t-1, \mathbf{x}t] + \mathbf{b}f) \\
\mathbf{i}t &= \sigma(\mathbf{W}i \cdot [\mathbf{h}t-1, \mathbf{x}t] + \mathbf{b}i) \\
\tilde{\mathbf{C}}t &= \tanh(\mathbf{W}c \cdot [\mathbf{h}t-1, \mathbf{x}t] + \mathbf{b}c) \\
\mathbf{C}t &= \mathbf{f}t * \mathbf{C}t-1 + \mathbf{i}t * \tilde{\mathbf{C}}t \\
\mathbf{o}t &= \sigma(\mathbf{W}o[\mathbf{h}t-1, \mathbf{x}t] + \mathbf{b}o) \\
\mathbf{h}t &= \mathbf{o}t * \tanh(\mathbf{C}t)
\end{aligned} \tag{2.27}$$

where $\mathbf{f}t$, $\mathbf{i}t$, and $\mathbf{o}t$ represent the forget, input, and output gates, respectively, $\mathbf{C}t$ is the cell state, and $\tilde{\mathbf{C}}t$ is a candidate for the new cell state.

Transformers. The introduction of Transformer architectures marked a substantial shift in the development of neural networks in NLP. Distinct from the sequential nature of RNNs and LSTMs, Transformers process entire sequences in a parallel manner. This is made possible through the use of an *attention mechanism*, introduced by Vaswani et al. [108], which influences the model's ability to capture contexts and relationships across various parts of the input sequence. The attention mechanism's primary function is to compute the relevance of different segments of the input sequence when analysing a specific part or word. It does this by generating attention scores that dictate the level of focus each component of a sequence gets:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.28}$$

where Q, K, and V denote query, key, and value vectors, respectively, and d_k is the dimension of the key vectors. The softmax function normalises these attention scores, effectively creating a weighted sum of the value vectors.

Transformers employ a unique structure comprising multiple encoders and decoders, each containing layers that operate identically. To compensate for the lack of sequential processing, Transformers incorporate positional encodings with input embeddings. These encodings inform the model with an awareness of the word order in the sequence. Furthermore, each layer in the encoder and decoder includes a fully connected feed-forward network, applied identically across all positions:

- The feed-forward network comprises two linear transformations, interspersed with a ReLU (Equation 2.4) activation function.
- This structure ensures that while the model processes data in parallel, it still main-

tains a comprehension of sequential relationships.

An important feature within layers is the multi-head attention mechanism, which facilitates simultaneous attention from different perspectives. The multi-head attention mechanism is defined by the authors [108] as follows:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \\ &\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \end{aligned} \tag{2.29}$$

Here, the model executes several attention operations in parallel (denoted as head_i), and their outputs are combined and linearly transformed. $\text{Concat}(\text{head}_1, \ldots, \text{head}_h)$ represents the concatenation of the outputs of individual attention operations (referred to as "heads"). Each head captures different aspects of the input data, where h denotes the total number of heads. The attention function itself calculates the scaled dot-product attention using projections of the query, key, and value where QW_i^Q, KW_i^K, and VW_i^V represent: the query matrix Q multiplied by a weight matrix W_i^Q specific to head i; the key metric K multiplied by its corresponding weight matrix W_i^K; and the value matrix V multiplied by its weight matrix W_i^V. W^O here denotes a weighting applied to the concatenated output of all of the heads, linearly transforming the concatenated vector to a defined output dimension.

BERT [26] represents a significant advancement in the application of Transformer architectures. Developed by researchers at Google, BERT's key innovation lies in its ability to process and understand the context of words in a sentence bidirectionally. BERT is solely based on the Transformer's encoder stack (no decoders, as used in the original Transformer model for translation tasks). The architecture is designed as a stack of identical layers, each layer consisting of multi-head self-attention mechanisms and fully connected feed-forward networks. Each layer processes the entire input sequence in parallel, significantly improving computational efficiency.

Traditional language models processed text in either left-to-right or right-to-left order. BERT, however, uses a mechanism called Masked Language Model (MLM) to understand the context of a word based on all of its surrounding words (both left and right context). In MLM, some percentage of the input tokens are randomly masked, and the model is trained to predict these masked tokens. This training forces BERT to develop a deep understanding of sentence context.

BERT is pre-trained on a large collection of text using two unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Unlike older, traditional language models that read text from left to right (or vice versa), BERT learns to understand the context of a word based its surrounding context. In MLM, some of the input tokens are randomly masked, and the model is then instructed to predict these masked

tokens. This training enables BERT to develop a deeper understanding of the context of sentences. For clarification, we define this mathematically.

Let's denote the masked token by $[MASK]$. If we have an input sentence with tokens $x_1, x_2, \ldots, x_N$ and we mask x_2, then the input becomes $x_1, [MASK], \ldots, x_N$. Mathematically, the loss function for MLM is:

$$\mathcal{L}_M LM = - \sum_{\text{masked} i} logP(x_i | x_1, \ldots, [MASK], \ldots, x_N). \tag{2.30}$$

Here, the sum is over the masked tokens, and $P(x_i)$ is the probability of the correct token x_i given its context, as predicted by BERT.

BERT also uses NSP for understanding the relationship between two sentences. It's a binary classification task to predict whether one sentence logically follows another. Given two sentences, A and B, BERT predicts the probability of B following A. Let $P(\text{IsNext}|A, B)$ represent the model's estimated probability that B is the next sentence following A, then the loss function for NSP is as follows:

$$\mathcal{L}_N SP = -logP(\text{IsNext}|A, B). \tag{2.31}$$

For specific tasks, BERT is fine-tuned with additional output layers. This involves minimal task-specific changes, preserving the pre-trained weights while adapting the model to specific NLP tasks.

2.4.2 Backpropagation

Backpropagation [92] is an algorithm used in the training process of neural networks, allowing them to learn from data and improve their performance. It is a method for optimising weights with respect to a loss function, updating the weights with a "backward pass" through the network. The process of backpropagation can be described in 5 steps: (1) the forward pass, (2) computing the loss, (3) the backward pass, (4) the chain rule of calculus, (5) and updating the weights.

In the forward pass, an input x is passed through the neural network to get the output. For each layer l in the network, the output $\mathbf{h}_l$ is calculated as:

$$\mathbf{h}_l = g_l(\mathbf{W}_l \mathbf{h}_{l-1} + \mathbf{b}_l) \tag{2.32}$$

where $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and biases of layer l, $\mathbf{h}_{l-1}$ is the output from the previous layer (with $\mathbf{h}_0 = x$), and g_l is the activation function for layer l.

At the output layer, the loss $\mathcal{L}$ is calculated:

$$\mathcal{L} = \text{Loss}(\mathbf{h}_{\text{output}}, \mathbf{y}) \tag{2.33}$$

where $\mathbf{h}_{\text{output}}$ is the output from the last layer of the network and y is the true label or value.

Backpropagation computes the gradients of the loss function with respect to each weight in the network, using the chain rule of calculus. For each layer l, starting from the output layer and moving backwards through the network, the gradient of the loss with respect to the weights $\mathbf{W}_l$ is calculated as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_l} = \frac{\partial \mathcal{L}}{\partial \mathbf{h}_l} \cdot \frac{\partial \mathbf{h}_l}{\partial \mathbf{W}_l} \tag{2.34}$$

where the term $\frac{\partial \mathcal{L}}{\partial \mathbf{h}_l}$, the gradient of the loss with respect to the output of layer l, is calculated using the gradients from either the subsequent layer or from the loss function for the output layer. The partial derivative $\frac{\partial \mathbf{h}_l}{\partial \mathbf{W}_l}$ depends on the specific activation function used in said layer.

Once the gradients are computed for all weights, the network's weights are updated using *gradient descent* [50]. Gradient descent is an iterative method used to minimise a loss function. The goal of gradient descent is to find the set of weights that minimises the loss, which often corresponds to finding the local minima of the loss function in the weight space.

In the context of a neural network, the loss function can be visualised as a multi-dimensional surface. Each point on this surface represents a particular set of weights and the corresponding value of the loss function. A local minimum is a point where the loss is lower than at all other points in the immediate vicinity. It's important to note that due to the complexity and high dimensionality of the weight space, there can be many such local minima. The objective of gradient descent is to navigate this surface and find the point (or set of weights) where the loss function reaches a local minimum. The basic update rule for a weight $\mathbf{W}_l$ is:

$$\mathbf{W}_l = \mathbf{W}_l - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}_l}. \tag{2.35}$$

Here, η is the learning rate, and $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_l}$ is the gradient of the loss function with respect to the weights. The gradient, a vector consisting of partial derivatives, points in the direction of the steepest ascent of the loss function. By moving in the opposite direction, the algorithm seeks to reduce the loss. This process of forward pass, backward pass, and weight update is repeated iteratively over multiple epochs (passes through the entire dataset), gradually reducing the loss and improving the model's predictions.

The learning rate, a small positive scalar determining the size of the weight update, plays a significant role in the convergence of gradient descent. A learning rate that's too high might cause the algorithm to overshoot the minimum, while a too low learning rate might result in a slow convergence or getting stuck in a local minimum. We make use of adaptive learning rate algorithms such as AdamW [54] to optimise the learning rate. In the next section, we will discuss other model optimisations used in this work.

2.4.3 Optimisation

Our work is considerably focused on maximising the efficiency of the transfer learning process. As such, we make use of a number of optimisation strategies in model training. These include arithmetic operation optimisations, and parameter-based optimisations, namely: the use of *Mixed Precision Training* to reduce model runtime; and the use of bottleneck *adapter* modules for efficient fine-tuning.

Mixed Precision Training. Mixed Precision Training [60] is an optimisation technique that uses both 32-bit (single-precision) and 16-bit (half-precision) floating-point types during neural network training. Half-precision (16-bit) floating points require half the memory compared to single-precision (32-bit). This reduction in memory usage allows for larger batch sizes or models to be trained on the same hardware. Many modern GPUs have specialised hardware for faster 16-bit arithmetic, which accelerates matrix multiplications and other operations common in neural networks.

Adapters. Adapters [42] are small trainable modules inserted between the layers of a pre-trained neural network. They are particularly useful for fine-tuning models on specific tasks without the need to retrain the entire network. Adapters consist of a down-projection layer, a non-linearity (like ReLU), and an up-projection layer. They are placed within the network and only the parameters of these adapters are trained, while the original pre-trained parameters are frozen.

Adapters are particularly useful in transfer learning due to the phenomenon of *catastrophic forgetting* [58, 84], where a neural network forgets previously learned information upon learning new tasks. Adapters mitigate this by allowing the network to maintain its original knowledge (encoded in the pre-trained weights) while adapting to new tasks through the trainable adapter layers. Given an input $\mathbf{h}$ to the adapter, the output $\mathbf{h}'$ is computed as:

$$\mathbf{h}' = \mathbf{h} + \mathbf{W}_d \cdot \mathrm{ReLU}(\mathbf{W}_u \cdot \mathbf{h} + \mathbf{b}_u) + \mathbf{b}_d \tag{2.36}$$

where $\mathbf{W}_u$ and $\mathbf{W}_d$ are the weights for the up-projection and down-projection layers,

respectively, and $\mathbf{b}_u$ and $\mathbf{b}_d$ are their biases.

2.5 Natural Language Processing

Natural Language Processing (NLP) integrates computer science, artificial intelligence, and linguistics to allow machines to interpret human language. Originating from linguistics [18] and machine translation [111] in the 1950s–60s, NLP has expanded to include speech recognition and text analysis tasks. It involves computational models for processing and generating human language. In the following sections, we provide a background on language representation for machine learning, from frequency-based measures to vector representations, providing a foundation for constructing representations from text. This thesis also explores various common tasks in NLP and IR, such as Topic Classification to classify documents into predefined categories; Part-of-Speech (POS) tagging, which assigns grammatical categories to words; Named Entity Recognition (NER), which identifies and categorises parts of text that mention, for example, names, locations, quantities; and Dependency Parsing, which analyses grammatical structures in documents.

2.5.1 Language Representation

Language representation is a key component of NLP. Traditional methods of representing language such as the *Bag-of-Words* [38] model and *Term Frequency-Inverse Document Frequency* [99] (TF-IDF), originating from Information Retrieval (IR), provide numerical representations of text based on their frequencies. However, these methods often overlook semantic relationships. Techniques based on *embeddings*, such as *Word2Vec* [61] and later *BERT* embeddings, map words or phrases from a vocabulary to vectors of real numbers in a continuous vector space, capturing semantic and syntactic relationships between words, as words with similar meanings tend to be closer in this high-dimensional space.

Bag-of-Words. The bag-of-words model is a simpler representation of text data that disregards grammar and word order but maintains the frequency of terms. It represents a document as a vector in a multidimensional space, where each dimension corresponds to a unique term in the corpus. If a term appears in the document, its value in the vector is its frequency in the document; if it does not appear, the value is zero. This can be formalised as:

$$BOW(d, T) = [f_1, f_2, \ldots, f_N] \tag{2.37}$$

where $BOW(d, T)$ represents the bag-of-words vector for document d over the set of all terms T, N is the number of unique terms in T, and f_i is the frequency of the i-th term

in d.

Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is a statistical measure, originating in IR, used to evaluate the importance of a word to a document in a corpus. The Term Frequency (TF) of a word in a document is the number of times the word appears in the document, normalised by the total number of words in the document. This is mathematically represented as:

$$tf(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d} \tag{2.38}$$

The Inverse Document Frequency (IDF), on the other hand, gauges the rarity of a term across the document corpus. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term. Mathematically, IDF is expressed as:

$$idf(t,D) = \log \frac{N}{d : t_i \in d} \tag{2.39}$$

where N is the total number of documents divided by the number of documents that contain the term, $d : t_i \in d$. Thus, the TF-IDF score of a term is the product of these two statistics:

$$tfidf(t,d,D) = tf_{t,d} \cdot idf_{t,D} \tag{2.40}$$

This score represents the relative importance of a term in the context of a particular document within the entire corpus, with higher values indicating greater importance.

Word2Vec: One prominent method of generating word embeddings is Word2Vec, developed by Mikolov et al. [61] Word2Vec represents words in a corpus by dense vectors which are learned by predicting a word's context (Continuous Bag of Words model, CBOW) or by using a word to predict its context (Skip-Gram model). In both models, context refers to the surrounding words within a certain window size. Word2Vec models use a shallow neural network architecture and are trained using techniques such as negative sampling, which enhances the efficiency of the learning process.

In the CBOW model, the embedding for a word is learned by using the embeddings of its context words. The objective is to maximise the probability of a target word given its context words. This is mathematically defined by the objective function:

$$\max \sum_{t=1}^{T} \log P(w_t | C_t) \tag{2.41}$$

where w_t is the target word, C_t is the set of context words, and T is the total number of training words. In the Skip-Gram model, the reverse is done: it predicts context words from the target word. The objective function for Skip-Gram is:

$$\max \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j}|w_t) \tag{2.42}$$

where w_{t+j} are the context words within the window $[-m, m]$ around w_t.

BERT-based Embeddings: The introduction of BERT [26] (Bidirectional Encoder Representations from Transformers) marked a significant evolution in the approach to generating word embeddings. We discuss the complexities of BERT's architecture in Section 2.4.1. BERT generates two types of embeddings: static and contextual. Static embeddings are pre-computed vector representations of words, obtained from the pre-training phase of the BERT model. These embeddings are the same for a given word regardless of its context in different sentences. They serve as a generalised representation of the word's meaning, derived from the model's exposure to vast amounts of text data during pre-training.

On the other hand, contextual embeddings are generated dynamically for each word based on its specific context within a sentence. This means that the same word can have different embeddings depending on its surrounding words, capturing nuances and varying meanings based on context. Contextual embeddings are computed by passing the entire sentence through the BERT model, which uses its multi-head self-attention mechanism to understand the relationship and dependencies of each word with respect to others in the sentence. This process is computationally more intensive as it requires a forward pass through the BERT model for each new sentence.

Contextual embeddings from BERT offer a more expressive representation of words. By considering the entire sentence, they capture not just the inherent meaning of a word but also how its meaning changes with context. However, the computational expense associated with generating these embeddings is a trade-off, as it requires processing power and time for each unique sentence, contrasting with the one-time computation of static embeddings.

2.5.2 Natural Language Processing Tasks

Natural Language Processing (NLP) encompasses a range of tasks aimed at enabling computers to understand, interpret, and generate human language. This section discusses a number of key NLP tasks and describes their objectives and mathematical formulations (where appropriate) in detail.

Topic Classification. Topic Classification involves categorising text into predefined top-

ics or labels. It's a supervised learning task where a model is trained on a labelled dataset. Given a document d and a set of possible topics $T = \{t_1, t_2, \ldots, t_n\}$ where n is the total number of topics, the task is to assign the most relevant topic to d. Often formulated as a classification problem, the output can be represented by either a Bernoulli or categorical distribution induced by, for example, a sigmoid or softmax function, respectively, over model outputs:

$$P(t|d) = g(\mathbf{W}_t \cdot \mathbf{h}_d + \mathbf{b}_t). \tag{2.43}$$

Here, $\mathbf{h}_d$ is a feature vector representing the document, $\mathbf{W}_t$, $\mathbf{b}_t$ are the weights and bias for the classification layer, and g represents the activation function.

Part-of-Speech Tagging. Part-of-Speech (POS) tagging is the process of identifying that a word corresponds to a particular part of speech by its definition and context. Using a model which captures sequences of text such as an RNN or LSTM (Section 2.4.1), the probability of a tag sequence $\mathbf{T}$ for a given word sequence $\mathbf{W}$ is modelled. The goal is to find the tag sequence that maximises the probability:

$$\mathbf{T}* = \arg\max_{T} P(\mathbf{T}|\mathbf{W}) \tag{2.44}$$

Named Entity Recognition. Named Entity Recognition (NER) is the task of classifying key information (entities) in text into predefined categories such as the names of organisations, locations, percentages, etc. Similar to POS tagging, NER is approached as a sequence labelling problem. For the sequence of words $\mathbf{W}$, the model outputs a sequence of labels $\mathbf{L}$ corresponding to entity types.

Dependency Parsing. Dependency Parsing identifies the grammatical structure of a sentence and establishes relationships between root words and words that modify those roots. Unlike POS tagging and NER, dependency parsers construct a directed graph representing the dependencies between words in a sentence.

2.6 Transfer Learning

In supervised learning, models are trained on a large corpus of labelled data, where the objective is to learn a mapping from inputs to outputs, based on the examples provided. However, this approach often encounters significant challenges, particularly when sufficient labelled data is not available. For instance, in less-resourced languages or specialised domains, the effort and cost to collect extensive labelled datasets can be impractical or prohibitively expensive. Additionally, the process of training models from scratch for

each new task demands substantial computational resources and time, which may not be feasible, especially in settings with limited budgets or computing capabilities.

These limitations highlight a gap in the traditional supervised learning approach. The inability to effectively learn from limited data and the high resource demand for training pose significant barriers to the development and application of models in a wide range of contexts. This gap requires an alternative approach that can circumvent these challenges while still leveraging the potential of machine learning models in processing and understanding natural language. Transfer learning emerges as a strategic solution to these challenges, leveraging existing knowledge and models to overcome the constraints of data scarcity and resource limitations.

Transfer learning is predicated on the idea that knowledge gained while solving one problem can be applied to a different but related problem. It typically involves using models that have been pretrained on large, diverse datasets. These models, having already learned a wide range of language features and patterns, can be fine-tuned with a smaller, task-specific dataset. The core of transfer learning is the transfer of knowledge from one task to another. This knowledge can be in the form of learned representations, features, or weights that are relevant and beneficial to the new task.

2.6.1 Definitions

In this section, we provide discuss and provide definitions of the numerous approaches to formulating transfer learning problems, primarily focusing on sequential transfer learning, the case used in our work. We begin with a formal definition of transfer learning, as proposed by Pan and Yang (2010) in their work *A Survey on Transfer Learning* [65]. Their analysis provides a foundational understanding of transfer learning and the different settings in which it is used. By contextualising their definition, we aim to align their theoretical framework with how transfer learning is used in our research.

In transfer learning, there exists a distinction between terms otherwise used interchangeably in machine learning literature, *domain* ($\mathcal{D}$) and *task* ($\mathcal{T}$). This distinction, as articulated by the authors, is not just terminological but reflects conceptual differences that are important for understanding the mechanics of transfer learning. In the simplest terms, a domain can be thought of as the context or environment in which a particular task is performed. Formally, it consists of two components, a feature space $\mathcal{X}$, which represents the set of characteristics the model considers, and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$, which describes how these features are distributed or how frequently they occur. A task, on the other hand, refers to the specific objective or problem the model is designed to solve. It is characterised by a label space $\mathcal{Y}$, which is the set of possible outputs, and a predictive function $f(.)$, which is the algorithm that maps inputs to predicted outputs. The definition of transfer learning is then:

Given a source domain $\mathcal{D}_S$ and its corresponding learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and its corresponding learning task $\mathcal{T}_T$, the goal of transfer learning is to improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

Transfer learning is typically organised into three separate categories: *Inductive*, *Transductive*, and *Unsupervised* transfer learning, each addressing different scenarios of domain and task configurations:

1. **Inductive Transfer Learning** is used when tasks between source and target domains may or may not be the same, but the domains are often identical. Formally, in inductive transfer learning, we consider a source domain $\mathcal{D}_S$ with its task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ with its task $\mathcal{T}_T$, where $\mathcal{T}_S \neq \mathcal{T}_T$, regardless of whether $\mathcal{D}_S = \mathcal{D}_T$ or $\mathcal{D}_S \neq \mathcal{D}_T$. It is further divided into two cases: (1) where there exists an abundance of labelled data in $\mathcal{D}_S$ (2) or where no labelled data in $\mathcal{D}_S$ is available.

2. **Transductive Transfer Learning** is applied when the source and target tasks are the same, $\mathcal{T}_S = \mathcal{T}_T$, but the domains are different, $\mathcal{D}_S \neq \mathcal{D}_T$, In this situation, there exists an abundance of labelled data in the source domain but no data in the target domain. Transductive transfer learning can also be further categorised by two cases: (1) the feature spaces are different between source and target domains, $\mathcal{X}_S \neq \mathcal{X}_T$, or (2) or the feature spaces are the same, $\mathcal{X}_S = \mathcal{X}_T$, but the marginal probability distributions are different, $P(X_S) \neq P(X_T)$.

3. In **Unsupervised Transfer Learning**, the target task, $\mathcal{T}_T$, is different but related to the source task, $\mathcal{T}_S$. However, in this setting, the focus is on solving unsupervised tasks in $\mathcal{D}_T$ such as clustering.

Our work closely resembles the transductive setting of transfer learning, where the domain we are transferring from contains a sufficient number of samples for transfer to a domain with no, few, or a limited number of samples. Additionally, we refer to the domain to transfer from as the *intermediate* domain, $\mathcal{D}_I$, signifying a two-step sequential transfer learning process. As mentioned above, transfer learning, in general, involves the process of transferring knowledge from one domain, typically known as the source, to another, referred to as the target. However, the concept of an "intermediate" domain introduces an additional step in this transfer process. It describes a sequential progression, where a model, often trained on a substantially larger corpus, is used a *base* model. The base model serves as the starting point for the sequential transfer process. The "intermediate" domain, then, represents a specific stage between the base model and the final target domain. In this framework, the base model is first fine-tuned on a task within the intermediate domain, designed to adapt to its characteristics. The base model acts as a rich repository of general

language understanding while the intermediate domain acts as a bridge, equipping the model with relevant features and knowledge that facilitate effective transfer to the target domain.

Throughout this work, the domains we transfer between are different, $\mathcal{D}_I \neq \mathcal{D}_T$, while the tasks—topic classification—are the same, $\mathcal{T}_S = \mathcal{T}_T$. To formalise this concept within the framework of transfer learning, we contextualise the above definition by Pan and Yang in our work as follows:

Given an intermediate domain $\mathcal{D}_I$ with its associated learning task $\mathcal{T}_I$, and a target domain $\mathcal{D}_T$ with its corresponding learning task $\mathcal{T}_T$, our objective is to enhance the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$. This enhancement is achieved through a sequential transfer process that begins with a base model M_B. Initially, M_B is fine-tuned on the intermediate task, resulting in a model MI, which is then further fine-tuned for the target domain and task, resulting in the final model $M_{I \to T}$.

2.6.2 Transferability Estimation

While transfer learning has proven to be an effective approach for leveraging knowledge across different domains and tasks, a critical question arises: *How do we determine which intermediate domain or task is most likely to yield the best transfer performance for a given target domain or task?* This question forms the basis of transferability estimation, a subfield of transfer learning that aims to predict the effectiveness of knowledge transfer between different domains or tasks. In the following section, we explore the methodologies and metrics used for estimating transferability, a central component of this thesis. We will explore the various factors that influence transferability, including the similarity between domains and the adaptability of the learned features.

Transferability Estimation refers to the assessment of the projected effectiveness of a model trained on one task when applied to a different task prior to training. Various approaches have been proposed to estimate transferability, ranging from using accuracy on a large-scale dataset like ImageNet as a proxy for transferability [47], to training different experts and selecting the best one based on metrics like image comparison, label matching, and performance [81].

Transferability is most commonly—though not exclusively [63, 114]—estimated by using statistical measures of *divergence*. Estimating divergence, in the context of machine learning, essentially measures the dissimilarity between data, often in the form of distributions or high-dimensional vectors. Kashyap et al. [46] provide a comprehensive taxonomy—which we extend in this work—of divergence measures used in different applications such as data selection, representation learning, and performance prediction. These divergence measures are used throughout different applications of transferability estimation. Poth

et al. [72] explores using different types of embeddings to estimate transferability, using divergence to compare domain representations. Plank and van Noord [68] use divergence to select training samples from data based on their divergence from a given test set, while Dai et al. [24] measure the overlap in terms to select data for pretraining language models. Ruder et al. [90] employ domain similarity metrics to select data for adaptation to new domains, using different (domain-, instance-, and subset-level) selection methods for inclusion.

Evaluating the performance and reliability of transferability estimation methods is as important as the estimation itself. One such evaluation metric is the Regret [86] metric, which measures the distance between the best source-target/intermediate-target combination found by a selection method and the optimal selection. The Regret metric provides a quantitative assessment of how well a transferability estimation approach can identify the most effective knowledge transfer configuration. Formally, it is defined as:

$$\text{Regret}_k = \frac{\overbrace{\max_{s \in \mathcal{S}} \mathbf{E}\left[T(s,t)\right]}^{O(\mathcal{S},t)} - \overbrace{\max_{\hat{s} \in \mathcal{S}_k} \mathbf{E}\left[T(\hat{s},t)\right]}^{M_k(\mathcal{S},t)}}{O(\mathcal{S},t)} \tag{2.45}$$

where $T(s,t)$ represents the performance on the target task t when transferring knowledge from an source/intermediate task s. $O(\mathcal{S},t)$ is the optimal expected performance on the target task when selecting the best source/intermediate task from the set $\mathcal{S}$, while $M_k(\mathcal{S},t)$ is the best performance one can achieve when selecting from a subset $\mathcal{S}_k$ of k tasks using a selection method. This metric allows us to quantitatively compare different task selection methods in terms of how close they come to an ideal selection. A lower regret value indicates that a selection method is closer to the optimal selection, implying a more effective method.

Chapter 3

A Framework for Transferability Estimation

3.1 Introduction

Intermediate task selection in transfer learning involves choosing tasks to train a model on before being fine-tuned for a specific target task. This approach is based on the idea that the knowledge gained from these intermediate tasks can improve the model's performance on the target task. However, there are several fundamental challenges associated with this process (see Section 1):

- **Identifying intermediate tasks that are relevant to the target task is challenging**. The knowledge required for good performance in the intermediate task should align with those needed for the target task. If the tasks are too dissimilar, the transfer of knowledge might not be effective. Moreover, not all knowledge learned in an intermediate task is transferable to a target task. Determining what knowledge is transferable and what is task-specific is a significant challenge.

- **Training intermediate tasks is expensive**. Exploring the space of intermediate tasks incurs substantial computational cost. This can be a limitation, especially when the number of candidate tasks for transfer is large.

- **The effectiveness of transfer is highly dependent on the quality of intermediate task data**. The effectiveness of intermediate task training is highly dependent on the availability and quality of data. Insufficient or noisy data, often found in natural language, can lead to suboptimal performance outcomes. Moreover, variations in data distribution between the intermediate and target tasks can affect the model's ability to generalise effectively. There is also a risk of negative transfer, where training on an intermediate task could potentially degrade the model's perfor-

mance on the target task. This often occurs when the intermediate task introduces irrelevant features that mislead the learning process.

In this chapter, we introduce a framework for estimating transferability to address the various risks and challenges in transfer learning. The core proposition of this framework is its ability to select intermediate tasks that are not only relevant but also beneficial to a specific target task. This selection process aims to significantly reduce resource consumption, including the associated environmental costs. In the following sections, we detail each component of this framework and explain how they collectively contribute to achieving these objectives.

1. In Section 3.2, we provide a high-level overview of the framework's architecture, describing how each component interacts with others to achieve a cohesive mechanism for intermediate task selection.

2. Section 3.3 describes our approach to resource tracking, where we implement a background process to monitor the efficiency of each component in terms of runtime, CO_2 emissions, and energy expenditure in kilowatt-hours.

3. Section 3.4 describes the specifics of our training and evaluating process in our *Domain Adapter Generation* and *Domain Transfer Analysis* components. We detail the specific experimental settings we explore to estimate the success of transfer learning in different scenarios.

4. In Section 3.5, we outline the *Representation Construction* component of our framework. The representations in this work include the two categories of representations we produce, derived from domain data: *Distributional* and *Embedding* representations. These representations encapsulate relevant features that can serve as indicators for transferability.

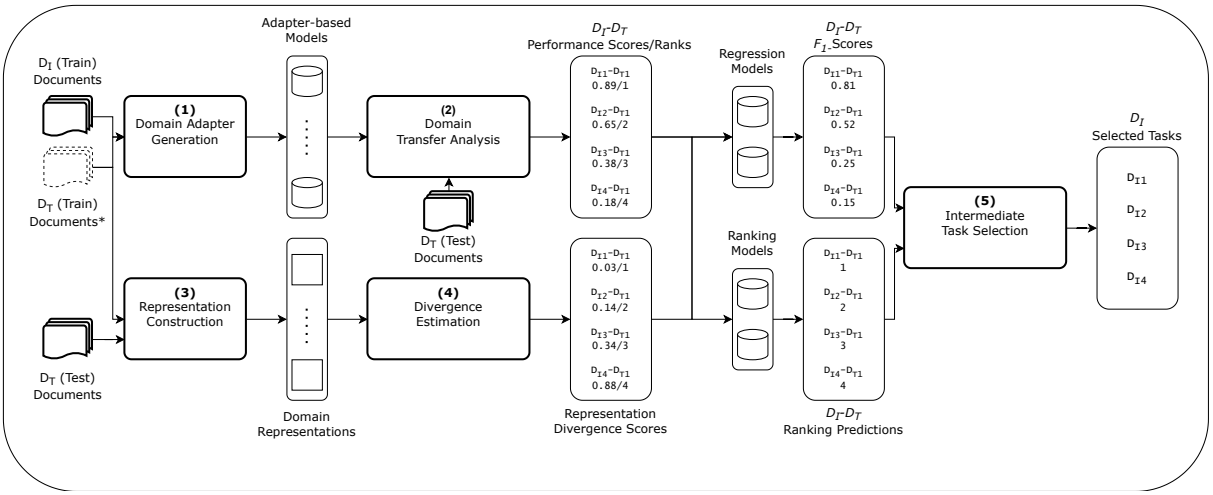
5. Section 3.6 describes the *Divergence Estimation* component. This section involves our process of computing the similarity between tasks using statistical measures of domain divergence. These measures allow us to quantify the relative distance between tasks, which then serve as features in subsequent task selection algorithms.

6. In Section 3.7, we discuss the core component of our framework for *Intermediate Task Selection*. This section assesses how effectively the divergence-based features can order intermediate tasks in terms of their transferability for a given target task, and discusses the trade-offs between performance and efficiency.

7. Lastly, in Section 3.8, we conclude by summarising the key aspects of our framework, encapsulating its overall contributions and implications.

3.2 Framework Overview

Figure 3.1: Overview of framework and the various stages therein. $\mathcal{D}_I$ and $\mathcal{D}_T$ refer to intermediate and target task, respectively. Regression and ranking models depend on the output of intermediate-to-target model evaluation and divergence estimation between their respective representations.



* Target domain training data, $\mathcal{D}_{T_{\text{Train}}}$, is only available in certain experimental settings.

Recall from Sections 1.2 and 1.4, the core objective and contribution of this thesis is to provide a comprehensive framework for transferability estimation. This involves: **(0)** tracking of the costs incurred at each stage of our framework, including representation construction and model training, and the evaluation of each; **(1)** training and fine-tuning models in a sequential transfer learning pipeline; **(2)** evaluating each of the models, where the performance of these models serves as our method of both evaluating representations and determining the ordering of intermediate tasks; **(3)** developing and evaluating representations that form the basis of our transferability estimation framework; **(4)** quantifying task relationships between the statistical divergence of representations derived from their data; and **(5)** predicting the ordering of intermediate tasks for a given target task, thereby providing a ranking of the most beneficial tasks for transfer. These concerns are divided into five individual components and a cost estimation background process that make up our framework, outlined as follows:

Cost Estimation. As a core contribution of our framework, we provide a comprehensive evaluation of the computational and environmental costs involved in each step. Processes within our framework are tracked using the *CodeCarbon*[1] Python package. Resource tracking runs as a background process within each stage of the pipeline tracked in terms of runtime (wall-clock time, provided by Python's `time` module), CO_2-equivalents, and kilowatt-hours of each process. The emissions and energy consumption of our processes

[1] `https://codecarbon.io/`

are determined by the hardware used throughout this work (outlined in Section 5.2.3).

(1) Domain Adapter Generation. This component involves training models used throughout our work using an adapter-based training approach [69]. Adapter modules, small bottleneck layers inserted within each transformer layer, allow for efficient model updates by freezing (meaning their weights are not updated during the backpropagation process) all other layers during training. Only the trained adapter layers and prediction heads need to be stored, leading to significant memory savings compared to full model fine-tuning, while maintaining comparable performance. Moreover, by only updating a small number of parameters during training, this mitigates the effects of catastrophic forgetting—a phenomenon where a model's previously learned knowledge is overwritten or "forgotten" when trained on a new task. Catastrophic forgetting is highly undesirable in transfer learning scenarios, as it can cause the model to lose its generalised capabilities acquired from pretraining or previous task training. The adapter approach mitigates this forgetting by leaving the majority of model parameters unchanged during task-specific training.

(2) Domain Transfer Analysis. In this stage, trained adapter models are subjected to a cross-domain transfer analysis. This involves evaluating either the intermediate model ($\mathcal{D}_I$ Model) or the intermediate-to-target model ($\mathcal{D}_I - \mathcal{D}_T$ Model) on target domain test data ($\mathcal{D}_{T_{\text{Test}}}$), where $\mathcal{D}_I \neq \mathcal{D}_T$. In the Zero-shot experimental setting, the intermediate model is directly tested on $\mathcal{D}_{T_{\text{Test}}}$ documents. Conversely, in Few-shot and Limited settings, the model undergoes additional fine-tuning with $\mathcal{D}_{T_{\text{Train}}}$ documents before its evaluation on $\mathcal{D}_{T_{\text{Test}}}$. Each model is designed for a binary classification task, and we use the binary F_1-score as the metric for evaluation. This approach essentially mirrors the traditional grid-search method of identifying optimal tasks. It involves experimenting with various task pair combinations to determine the one that yields the best performance in terms of transfer learning.

(3) Representation Construction. This stage involves the construction of representations derived from domain data. These representations serve as a method of encapsulating domain characteristics and is divided into two distinct categories: (1) *Distributional Representations* (Chapter 6) and (2) *Embedding Representations* (Chapter 7). We estimate the statistical similarity (or divergence) between these representations in the next stage of our framework.

(4) Divergence Estimation. This stage involves estimating the divergence between constructed representations using statistical measures in order to: (1) measure the projected effectiveness of our representations, and (2) use the divergence between intermediate and target pairs as features in our task. In transferability estimation, the closer—or less

divergent—domains are, the more likely they are to result in improved performance over a more distant—or more divergent—pair of tasks. The outputs of these divergence measures serve as input features to the models used in our final component.

(5) Intermediate Task Selection. The final component uses the divergence between task representations to recommend an ordered sequence of intermediate tasks to the user. Akin to a recommendation system, we aim to suggest intermediate models that, when used for transfer learning to the target task, will likely yield optimal performance. We evaluate the robustness of our approach through three criteria: the achieved target task performance, computational efficiency, and the balance/trade-off between the two.

3.3 Cost Estimation

We track and report the following: (1) training and inference times for intermediate, target, and intermediate-to-target models; (2) representation construction times, including additional costs incurred by representation-specific processes (e.g. vocabulary construction); and (3) divergence estimation, including costs incurred by more resource-intensive divergence estimation approaches (e.g. instance-based comparisons).

We report the averaged costs incurred to use our framework for a single target domain, including costs associated with constructing representations for both the target domain and the costs of constructing representations for candidate intermediate tasks.

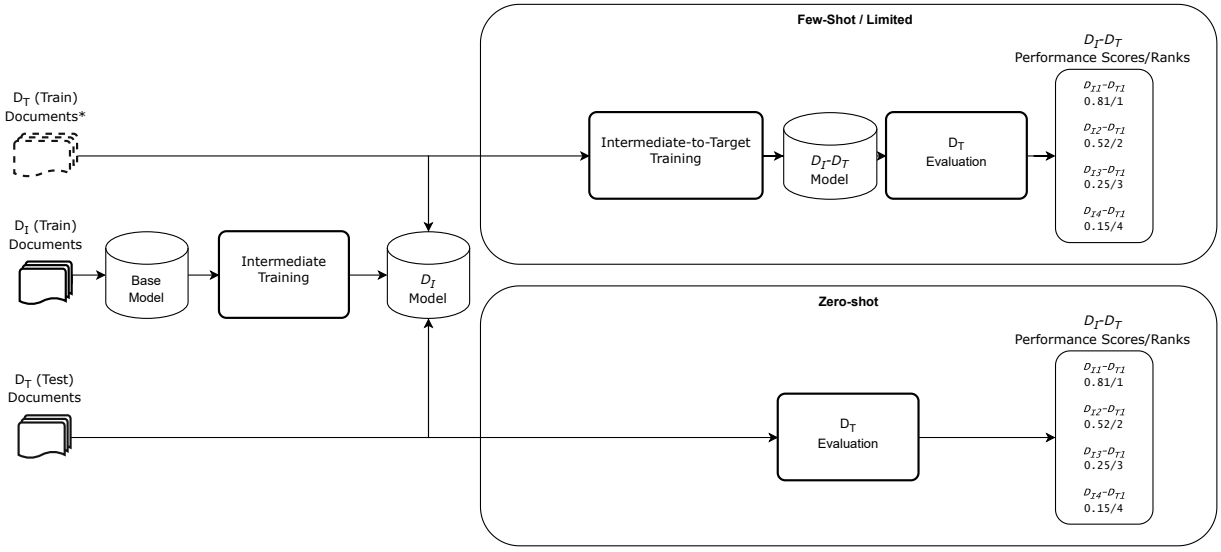
3.4 Domain Adapter Generation/Transfer Analysis

In our work, which focuses on sequential transfer learning, adapter-based training presents a distinct advantage in mitigating catastrophic forgetting. By freezing the non-adapter layers, the core knowledge embedded in the pretrained model can be preserved, thus preventing the overwriting of previously learned information. This is particularly important in sequential transfer learning, in which the model is repeatedly trained on a new task. Adapter layers allow the model to acquire new capabilities without losing or substantially altering the general knowledge information contained in the original model. In this stage of our framework, we train adapters on intermediate ($\mathcal{D}_I$) tasks and, if target domain training data ($\mathcal{D}_{T_{\text{Train}}}$) is available, further fine-tune on target tasks in a sequential fashion:

Intermediate ($\mathcal{D}_I$) Training. In this phase, we integrate bottleneck adapters into a pretrained model. These adapters are trained using the full-sized intermediate task dataset. After training, the resulting $\mathcal{D}_I$ Model is saved to disk.

Intermediate-to-Target ($\mathcal{D}_I - \mathcal{D}_T$) Training. When target domain training data is

Figure 3.2: Stages for the *Domain Adapter Generation* and *Domain Transfer Analysis* components of our framework. $\mathcal{D}_I$ and $\mathcal{D}_T$ refer to intermediate and target tasks, respectively. Models are trained using adapters, in a sequential fashion. $\mathcal{D}_I \rightarrow \mathcal{D}_T$ performance scores serve as ground truth in downstream experiments.



* Target domain training data, $\mathcal{D}_{T_{\text{Train}}}$, is only available in certain experimental settings.

accessible (in Few-shot or Limited scenarios), we use the adapter trained on the intermediate task. This adapter is then fine-tuned using the target domain training data. We use two types of target domain training data: one with a few samples and another with a limited but adequate number of samples. For each experimental setting, this process creates distinct $\mathcal{D}_I - \mathcal{D}_T$ models, which are later used for evaluation with the target test data ($\mathcal{D}_{T_{\text{Test}}}$).

Target Domain Evaluation. We evaluate the target domain test data ($\mathcal{D}_{T_{\text{Test}}}$ Documents) using either the $\mathcal{D}_I$ Model or the $\mathcal{D}_I - \mathcal{D}_T$ Model. The $\mathcal{D}_I$ Model is used in zero-shot settings, where no training data for the target domain is available. The $\mathcal{D}_I - \mathcal{D}_T$ Model is used when there are few or limited target domain training samples available, allowing further fine-tuning to adapt the model to the target domain. As mentioned previously, since each class if formulated as a binary classification task, we evaluate using binary F_1-score.

3.5 Representation Construction

The next component of our framework concerns the creation of effective domain representations. The core premise here is that the power of any divergence measure hinges on the quality of domain representations, i.e. their ability to capture the salient characteristics of the underlying domain. In this thesis, we focus on constructing two distinct types of domain representations: *Distributional Representations* and *Embedding Representations*.

Distributional representations such as Term Distributions (TD), Linguistic Feature

Table 3.1: Overview of the representations used throughout this work.

Category	Representation	Abbreviation
Distributional Representations	Term Distributions	TD
	Linguistic Feature Distributions	LFD
	Topic Frequency Distributions	KFD
Embedding Representations	BERT Static/Contextual	BERT (S/C)
	Probability-Weighted Embeddings	PWE
	Distributive Contextual Embeddings	DCE

Distributions (LFD), and Topic Frequency Distributions (KFD) focus on the frequency and occurrence of features within the data. This approach is effective for identifying common patterns and general trends, offering a clear overview of the domain's characteristics. The main advantage is its simplicity and ease of interpretation. However, distributional representations struggle to capture more complex, contextual relationships within data.

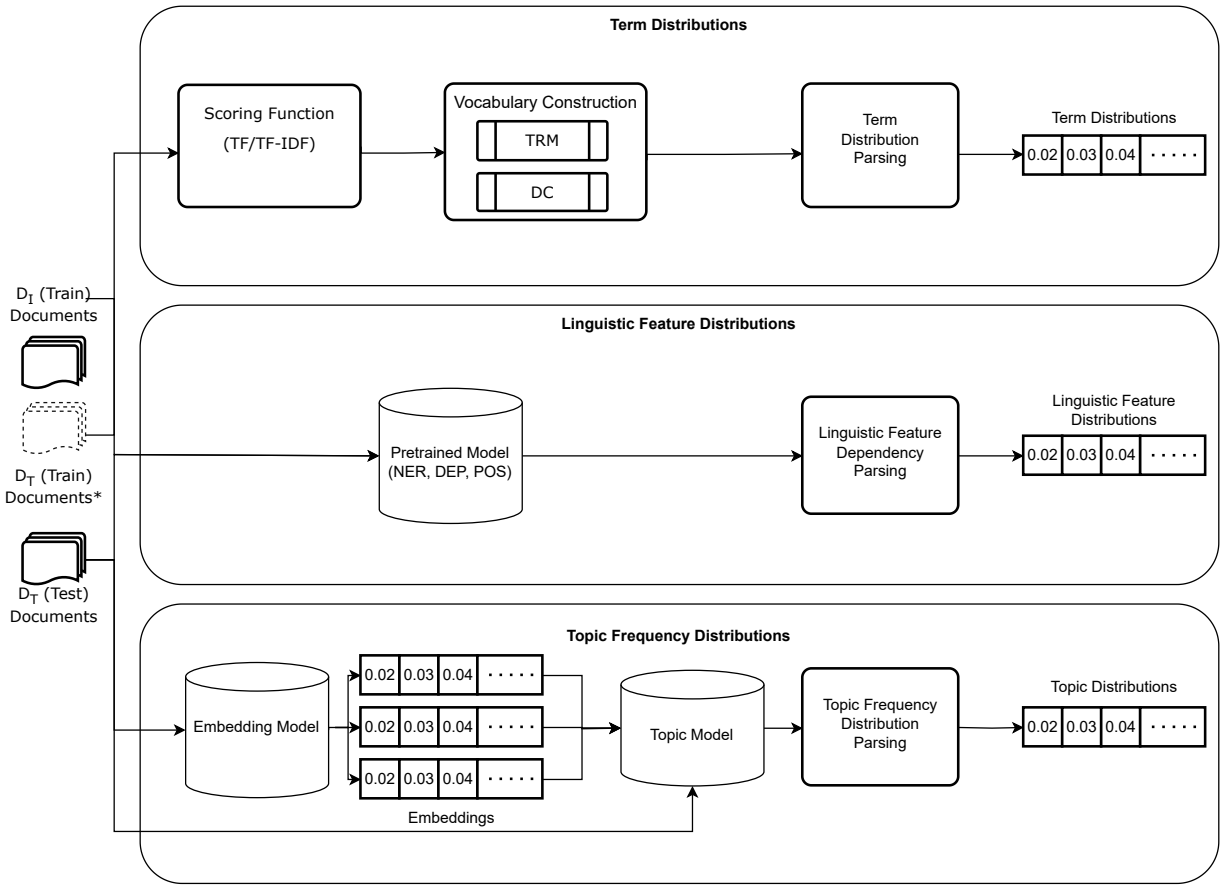
Embedding representations such as static/contextual BERT Embeddings (BERT (S/C)), Probability-Weighted Embeddings (PWE), and Distributive Contextual Embeddings (DCE) are high-dimensional vectors that encapsulate semantic and syntactic information and are adept at encoding complex relationships and contextual meanings. While they offer a richer analysis, they are often more computationally expensive to produce.

3.5.1 Distributional Representations

This category captures the frequency or occurrence patterns of linguistic elements within the dataset, such as term frequencies or linguistic feature distributions. By considering the distributional nature of words or features across documents, this representation type provides a comprehensive view of the textual data, preserving the inherent relationships and structures present in the corpus. In this work, we cover the following distributional representations: Term Distributions, Linguistic Feature Distributions, and Topic Frequency Distributions.

Term Distributions (TD). This category includes Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) distributions. Each item in a term distribution represents a unique term. For each target domain, we build a fixed-size vocabulary from which we construct term distributions. We approach vocabulary construction by considering the: (1) *Term Ranking Method* (TRM), where terms are ranked by either their frequency or TF-IDF score to determine which terms are to be included in the vocabulary; and the (2) *Domain Context* (DC), where vocabularies are either constructed solely from the target domain or from the target domain and all available intermediate domains (maintaining $\mathcal{D}_I \neq \mathcal{D}_T$).

Figure 3.3: Overview of the Distributional Representation Construction Process. The diagram illustrates the transformation of text-based datasets into three main categories of representations: Term Distributions, Linguistic Feature Distributions, and Topic Frequency Distributions. $\mathcal{D}_I$ and $\mathcal{D}_T$ refer to intermediate and target tasks, respectively. For Term Distributions, TRM and DC refer to the Term Ranking Method and Domain Context vocabulary construction methods, respectively. For Linguistic Feature Distributions, NER, DEP, POS refer to named entities, linguistic dependencies, and part-of-speech tags.



* Target domain training data, $\mathcal{D}_{T_{\text{Train}}}$, is only available in certain experimental settings.

Linguistic Feature Distributions (LFD). In this category of representations, we consider grammatical and semantic structures within each corpus. We focus on (1) part-of-speech (POS) tags such as nouns, verbs, and adjectives; (2) named entities (NE) such as persons, organisations, and locations; and (3) syntactic relationships and linguistic dependencies (DEP) between terms. To identify these structures within text, we make use of pretrained models from the Python-based spaCy library. In particular, we use the *Morphologizer*[2] for part-of-speech tagging, the *EntityRecognizer*[3] for identifying named entities, and the *DependencyParser*[4] for identifying linguistic dependencies. We count the number of these attributes across each corpus to produce POS, NE, and DEP frequency distributions.

[2] https://spacy.io/api/morphologizer
[3] https://spacy.io/api/entityrecognizer
[4] https://spacy.io/api/dependencyparser

Topic Frequency Distributions (KFD). Topic modelling is a method used to uncover hidden thematic structures within large collections of text. It identifies topics, which are essentially clusters of words, that frequently occur together across the documents. This process involves: (1) training the topic model on all available intermediate domain training data, and if available, on the target domain training data as well, enabling the model to learn and identify distinct topics that are prevalent in these domains; (2) transforming the documents of each domain into a distribution of document-topic cluster allocations where each document is represented by an integer corresponding to the cluster it has been assigned to; and (3) aggregating these document-topic distributions into a one-dimensional view (the KFD), where each item represents a cluster in the one-dimensional distribution, whose value is the normalised frequency of documents in that cluster.

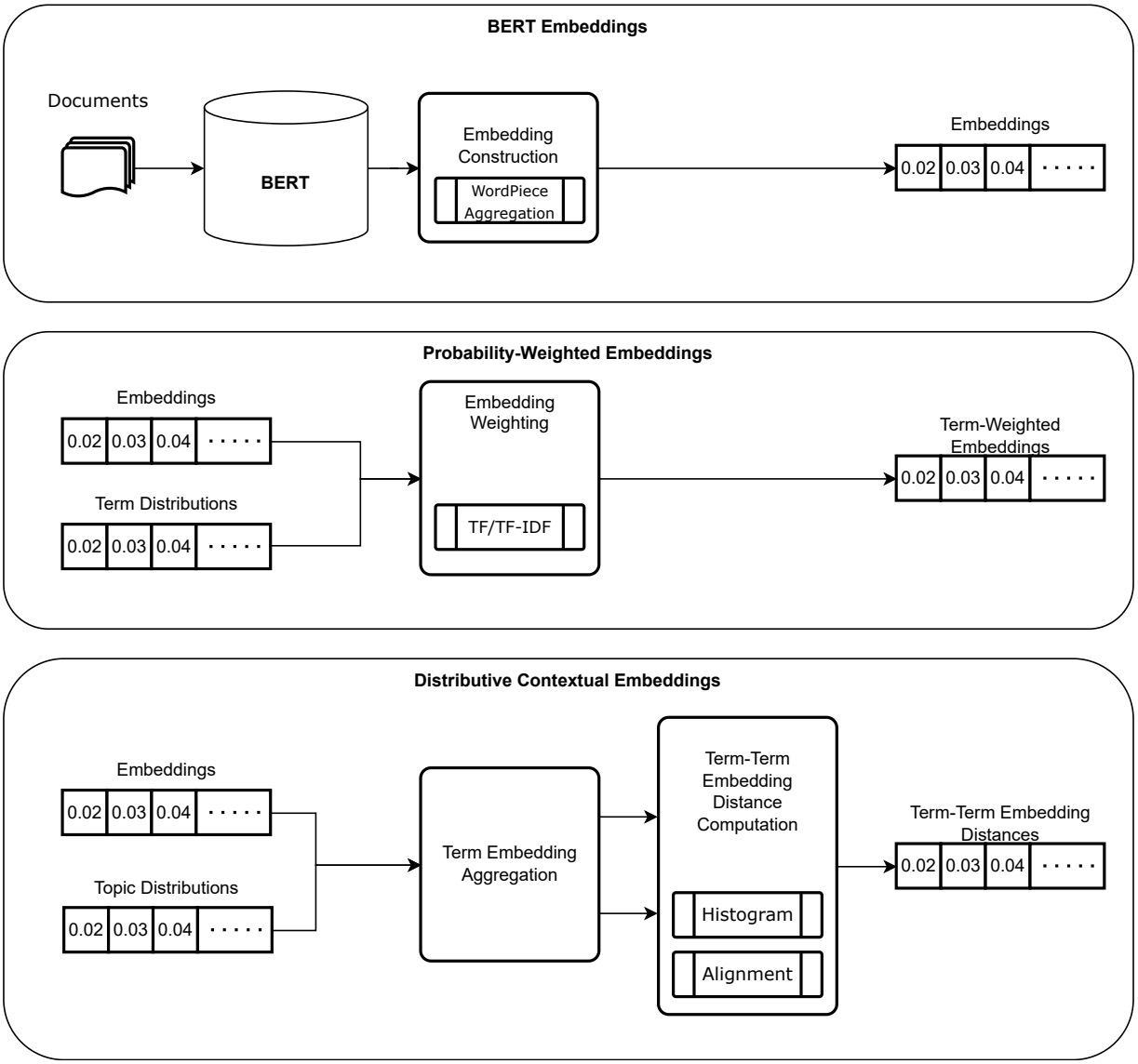
3.5.2 Embedding Representations

Embedding Representations are designed to capture the complex semantic and syntactic characteristics of language within the dataset. They achieve this by mapping terms to high-dimensional vector spaces. In these spaces, the positioning of each term reflects its contextual relationships and meanings in the dataset. This approach allows for capturing complex relationships between terms or phrases in a corpus. In our work, we explore three types of Embedding Representations: BERT Embeddings, Probability-Weighted Embeddings, and Distributive Contextual Embeddings.

BERT Embeddings (BERT). In this process, we use embeddings from a pretrained BERT model, extracting them from either the word embedding layer or the second-to-last layer. The word embedding layer provides general-purpose term vectors, offering a fast and efficient way to capture term representations. Contextual term vectors, obtained after processing documents through the model, provide a deeper, context-specific representation of terms. This dual approach allows for flexibility in choosing between quick, general-purpose embeddings and richer, context-specific ones.

Probability-Weighted Embeddings (PWE). This method combines the Term Distributions (and their vocabularies) from the previous chapter with BERT embeddings. We map terms that are in-vocabulary to their respective domain-specific term distributions, obtaining frequencies for each term in different domains. These frequencies are then used to modify the extracted BERT term vectors. Specifically, we amplify these vectors by multiplying them with their associated probabilities, incremented by one to preserve the original information in the embeddings. This technique enhances the BERT embeddings with domain-specific frequency information, enriching the representation with both semantic context and frequency-based importance.

Figure 3.4: Overview of the Embedding Representation Construction Process. The diagram illustrates the transformation of text-based datasets into three main categories of representations: BERT Embeddings, Probability-Weighted Embeddings, and Distributive Contextual Embeddings.



Distributive Contextual Embeddings (DCE). This approach synthesises the theoretical bases of both distributional and embedding representations. To create distributive contextual embeddings, we aggregate contextual term vectors within each domain, focusing on those terms that are part of domain-specific vocabularies. This aggregation results in a distribution of averaged term vectors, where each vector encapsulates the contexts in which the term is commonly used within the domain.

3.6 Divergence Estimation

Divergence Estimation is a process where statistical measures are used to gauge the similarity or dissimilarity between distributions or vector representations like embeddings. In this thesis, these measures are used to determine the relatedness between task pairs, inferred from the divergence of their respective representations. Drawing inspiration from previous studies, we have developed a taxonomy of divergence measures, as detailed in Chapter 4, extended from and inspired by a taxonomy created by Ramesh Kashyap et al. [46]. This taxonomy includes five categories of divergence measures, each distinct in their characteristics and applications:

Geometric. These measures calculate the direct distance between data points in a vector space. We use Cosine, L_1, and L_2 distances. Cosine distance is particularly effective when the magnitude of vectors is not as important as their orientation or direction, making it ideal for text similarity where frequency counts are normalised. L_1 and L_2 distances, on the other hand, are sensitive to the magnitude of differences in term frequencies, suitable for tasks where absolute differences in term usage are significant.

Information-theoretic. These measures assess the divergence between probability distributions of terms. We use Jensen-Shannon and Rényi divergences, and Bhattacharyya distance. The Jensen–Shannon divergence is symmetric, meaning that the divergence from an intermediate to a target distribution is the equal to the divergence from a target to an intermediate distribution. This symmetric property makes it particularly reliable for estimating the representational divergence between domains, as the direction of the divergence calculation does not affect the result when comparing intermediate and target task distributions. Rényi divergence introduces a parameter to adjust the sensitivity. For example, when comparing term frequencies, this flexibility allows a certain degree of control over the amount of weight given to large or small differences in term occurrence. Bhattacharyya distance distance can be used to assess the extent to which the vocabulary of two text corpora overlap, where a lower distance indicates a higher degree of overlap, meaning that the two texts share a large number of common terms.

Optimal Transport. These focus on the cost of transforming one distribution into another. We use Wasserstein-1 and Wasserstein-2. Wasserstein-1, or Earth Mover's Distance, measures the minimal effort needed to align one distribution with another and is useful when dealing with distributions that have minor shifts or differences. Wasserstein-2, being quadratic, is more sensitive to larger shifts in distributions, making it appropriate for scenarios where major differences in term distributions are being compared.

Statistical Moments. These measures look at the characteristics of distributions, such

as their shape and spread. Our chosen measures are CORAL, Central Moment Discrepancy, and Maximum Mean Discrepancy (with various kernels). CORAL is effective in comparing the covariance of two distributions, useful in assessing the structural similarity. Central Moment Discrepancy can capture higher-order moments (like skewness and kurtosis), valuable in detailed distribution comparison. Maximum Mean Discrepancy, with its flexibility in kernel choice, can adapt to various types of distributional differences, from subtle to pronounced. We also use mean, variance, skewness, and kurtosis to describe the central tendency, spread, the *tailedness* or heaviness of distributional "tails", and the *peakedness* or the height/density of each tail relative to a normal distribution.

Diversity Measures. These quantify the variety within a distribution. We use Shannon and Rényi entropies and Simpson's Index. Shannon entropy is useful for measuring the unpredictability or randomness in term usage, helpful in analysing text complexity. Rényi entropy extends this by offering a parameterised approach to diversity measurement. Simpson's Index, on the other hand, is adept at quantifying the diversity or uniformity in term usage.

3.6.1 Aggregating Representations

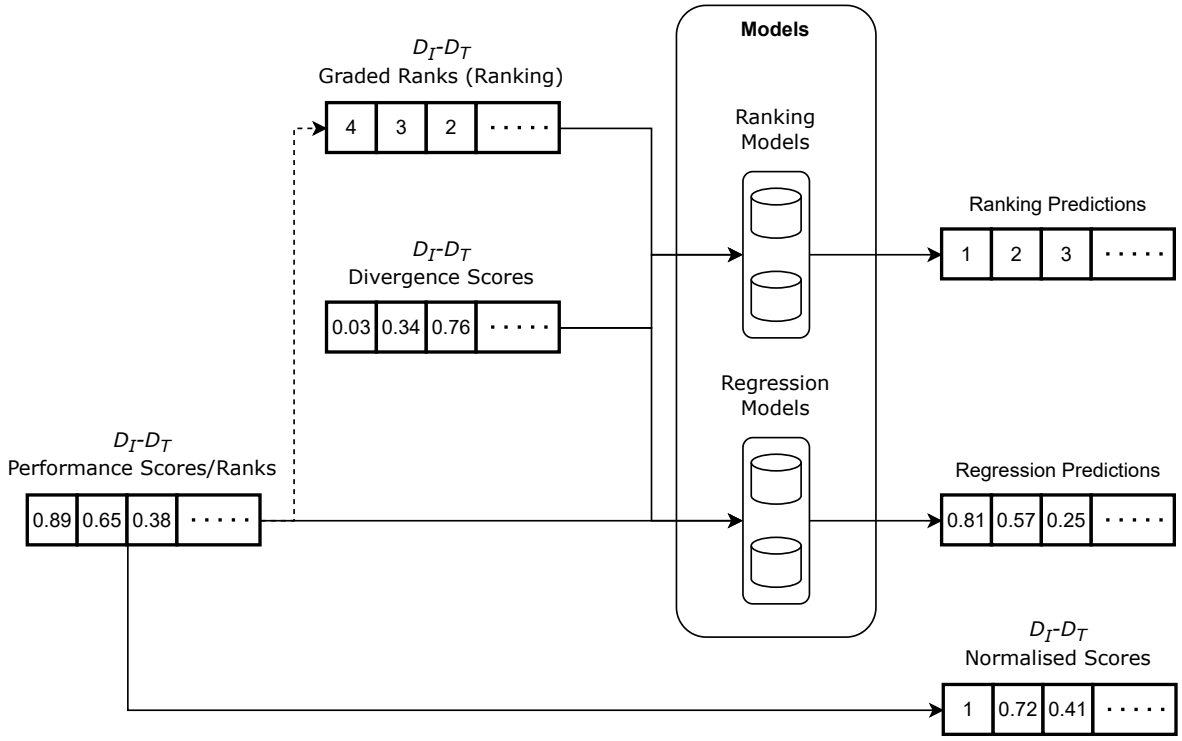
Depending on the representation, it may be more effective to calculate the divergence between domains at the document-, subset-, or domain-level. Hence, in this stage, we include a component to assess the effects of aggregating representations, allowing us to calculate the divergence between representation pairs at different levels of abstraction. In this thesis, we refer to: (1) document-level comparisons as *instance*-based aggregation; (2) subset-level comparisons—where frequency-based data in multiple documents is aggregated into a single representation—as *centroid*-based aggregation; and (3) corpus-level comparisons—where frequency-based data across every document in the corpus is consolidated into a single representation—as *domain*-based aggregation.

In choosing the appropriate level of aggregation for divergence calculation, we aim to investigate whether we can better quantify the divergence between domains through a more focused, document-level analysis or a more cost-effective aggregation approach that captures general characteristics.

3.7 Intermediate Task Selection

In selecting intermediate tasks for transfer, we make use of four different regression and ranking models: Linear Regression, XGBRegressor [17], LambdaRank [14], and LambdaMART [15]. The rationale behind this selection of models is twofold. Our ground truth for model performance is based on the F_1 score of the positive class. Intuitively, we be-

Figure 3.5: Ranking/Regression pipeline. Input rank assessments (Graded Ranks) are computed from performance scores and correspond to our scoring system defined in Section 8.2.3. Ground truth performance scores are normalised ($D_I - D_T$ Normalised Scores) when used with NDCG.



gin with a regression analysis to investigate the efficacy of domain divergence in directly estimating these performance scores.

For regression, we opted for Linear Regression and the XGBRegressor. Linear Regression serves as a baseline due to its simplicity and interpretability, providing insights into the linear relationships between feature vectors and target performance. On the other hand, XGBRegressor, an implementation using gradient-boosted trees, was chosen to capture potential non-linear relationships in the data, allowing for more intricate modelling.

Regression models allow for an estimation of task performance, however, the nature of our problem is inherently about ranking task pairs based on their suitability for transfer. As such, we include two ranking models in our experiments: LambdaRank and LambdaMART. LambdaRank utilises a pairwise comparison approach, focusing on the relative order between pairs of items, LambdaMART employs a listwise methodology. By considering the entire list of intermediate task candidates, LambdaMART optimises the overall ranking, potentially capturing relationships between tasks that LambdaRank may be unable to.

Figure 3.5 visualises the pipeline of task selection component. In training our models for intermediate task selection, every possible intermediate-to-target pairing across datasets is considered. For evaluations concerning a particular target domain, pairs tied to that

domain were designated as the test set, while all other pairs constituted our training set. To prevent inadvertent exposure to the domain under assessment, any instances where $\mathcal{D}_I = \mathcal{D}_T$ were excluded from the training set.

Regression. In regression tasks, we aim to directly predict the F_1-score of each intermediate-to-target ($\mathcal{D}_I - \mathcal{D}_T$) model. Typically, the output predictions of regression models are evaluated by measures such as *Root Mean Squared Error* (RMSE), however, since our objective is to recommend optimal task combinations, we are primarily concerned with the relative ranking inherent to the ordering of these predicted scores. Hence, we evaluate the outputs of our regression models using ranking metrics such as *Normalised Discounted Cumulative Gain* (NDCG) (Section 2.3.3) to determine if the ordering of our predictions is close to the ordering of ground truth performance scores. In some experimental scenarios, the variation in ground truth scores may be minimal. This may be a problem when applying the NDCG metric, since small differences in these scores could lead to disproportionate and overly optimistic evaluations of model ranking. To solve this problem, we normalise our ground truth performance scores to a range between 0 and 1, where 1 represents the best model performance. This normalisation process ensures a more balanced and realistic comparison as it increases the relative differences in the performance of the model, which makes it easier to distinguish and accurately rank models in terms of effectiveness.

Ranking. In ranking tasks like evaluating model performance, directly using the performance scores can be misleading. This is because the scores are often very close to each other, making it hard to tell which model is truly better. Using a ground truth based directly on model performance scores might result in a ranking that doesn't accurately reflect the differences between the models. To solve this, we implement a "grading system", grouping scores into categories or grades that represent different levels of relative performance. This is similar to how, in search engine ranking, web pages are ranked based on their relevance to the search query. The highest grade (4) is for the best-performing models; a grade of 3 is assigned to models that are among the best-performing, but not the best (ranks 2-3); and a grade 2 is for models that are considered to be high-performing, but not as good as models in the top 3 ranks (ranks 4-5). Models that perform better than the bottom 30% but are not in the top ranks get a grade of 1. The lowest performers, in the bottom 30%, get a grade of 0. This grading makes it easier to see which models are doing well and which are not, and more importantly, to signal to the model which combinations of tasks to recommend and which to not.

3.8 Conclusions

In this section, we proposed a novel framework for transferability estimation for natural language processing tasks. This framework is comprised of five components (along with a cost estimation background process). For each component, we have described the functionality of that component and provided a rationale behind its design. In particular, in Section 3.3, we described the resource-tracking component which allows us to track the end-to-end runtime, the CO_2 emissions, and the energy expenditure of each process in the framework. In Section 3.4, we described our process to training and evaluating adapter-based modules for transfer learning, providing a resource-efficient alternative to fine-tuning a large number of models and methods to evaluate them. In Section 3.5, we described the representations used in this work and the benefits and drawbacks of each, along with describing how we compute the statistical similarity between pairs of these representations in Section 3.6. Lastly, in Section 3.7, we describe our process for intermediate task selection for transfer learning and how we approach the problem from both a regression and ranking perspective, ultimately combining the outputs from previous components to provide a comprehensive framework for this task. In the next chapter, we will describe our experimental setup and the foundational knowledge required to understand our approaches in later experimental chapters. This chapter includes the rationale behind our choice of datasets, our approach to transforming these datasets for use in our framework, various model optimisation strategies, and tables of performance scores of our transfer learning models that serve to motivate subsequent chapters.

Chapter 4

Taxonomy of Divergence and Diversity Measures

In this chapter, we present an expanded taxonomy of divergence measures, building upon the foundational classifications introduced in prior research. The original taxonomy, defined in the works of Ramesh Kashyap et al.. [46], categorise divergence measures into three primary groups: *Geometric*, *Information-theoretic*, and *Higher-order*. In our extended taxonomy, we retain the first two categories, while redefining the *Higher-order* category as *Statistical Moments*. This redefined category now includes both the higher-order measures defined in the original taxonomy–Central Moment Discrepancy, Maximum Mean Discrepancy, and CORAL—and includes the first four central moments—mean, variance, skewness, and kurtosis—which are used to statistically describe individual distributions. Additionally, we integrate insights from the work of Ruder and Plank [91], introducing a new category titled *Diversity Measures*, which incorporates measures Entropy, Rényi Entropy, and Simpson's Index, explored in their research. Furthermore, we propose the inclusion of *Optimal Transport* as a new category, which features Wasserstein-1 and its quadratic case, Wasserstein-2. The following sections will describe each of these respective categories and provide formal mathematical definitions and descriptions of each method.

4.1 Geometric Measures

Geometric measures can be defined as measures which calculate the distance between pairs of continuous representations, such as between vectors in a metric space. In our work, we use them with both embedding-based vector representations and distribution-based representations, treating the latter as vectors.

Cosine Similarity is a measure that quantifies the similarity between vectors. Mathematically, it is the division between the dot product of the vectors and the product of the

Euclidean norms of each vector. In order to maintain uniformity with the rest of the divergence measures in this work, we compute the **Cosine Distance** using $1 - cos$. Let $\vec{p}$ and $\vec{q}$ be two vectors in $\mathbb{R}^n$, then:

$$cos(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{\|\vec{p}\| \cdot \|\vec{q}\|} \tag{4.1}$$

l_p-norm: l_1 (i.e., **Manhattan distance**) and l_2 (i.e., **Euclidean distance**) distance both belong to a broader family of distance measures known as "p-norms". **Euclidean** distance can be described as the length of a line segment between two points in Euclidean space. Manhattan distance, on the other hand, is the distance between two points, computed as the sum of the absolute differences of their Cartesian coordinates. Thus, if we consider two vectors p and q, then the Euclidean distance, d_2, and the Manhattan distance, d_1, between them is given by:

$$d_2(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{4.2}$$

$$d_1(p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{4.3}$$

4.2 Information-theoretic Measures

Information-theoretic measures calculate distances or similarities between pairs of probability distributions. Representations such as n-gram distributions over a corpus can be used with f- (where f is convex) or α-divergences (parameterised by a real value α where $\alpha \geq 0$ and $\alpha \neq 1$).

Kullback-Leibler (KL) Divergence [48] is a statistical distance measure between probability distributions. Consider two probability distributions, P and Q. Here, P represents the data or observations and Q—known as the reference probability distribution—represents some theoretical model we are comparing to. KL Divergence can then be described as how well P approximates Q, that is, the divergence between P and Q:

$$D_{KL}(P||Q) = \sum_{x} P(x) ln \left(\frac{P(x)}{Q(x)} \right) \tag{4.4}$$

If for any event, x, the probability according to Q (i.e. $Q(x)$) is equal to zero, then for KL Divergence to be applicable, P must also be zero for the same event (i.e. $P(x) = 0$). If, however, $Q(x) = 0$ and $P(x) > 0$, then KL Divergence is undefined, as it fails to handle such discrepancies. For this reason (and since KL divergence is non-symmetric, i.e.

$D_{KL}(P||Q) \neq D_{KL}(Q||P)$), we do not use KL divergence in this work. However, it forms the basis for other divergence measures used in this thesis.

Jensen-Shannon Divergence [52] can be considered an extension of the Kullback-Leibler divergence:

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \tag{4.5}$$

where $M = \frac{1}{2}(P+Q)$. Unlike KL-Divergence, Jensen-Shannon is symmetric, i.e. $D_{JS}(P||Q) = D_{JS}(Q||P)$. This means that it is suitable for use as a distance metric, since the reverse of the inputs (swapping P and Q) results in the same output.

Rényi Divergence [87] is an α-divergence, a measure of dissimilarity between probability distributions parameterised by α where $\alpha \neq 1$. This parameter makes Rényi divergence flexible in that different values of α can emphasise different aspects of inputs P and Q. We can define Rényi divergence as:

$$D_{\alpha}(P||Q) = \frac{1}{\alpha - 1} log \left(\sum_{x} \frac{P(x)^{\alpha}}{Q(x)^{\alpha-1}} \right) \tag{4.6}$$

By changing α, one can use the Rényi divergence to interpolate between different divergence measures; for example, when $\alpha = \frac{1}{2}$ it equals the Bhattacharyya distance (Equation 4.7). Higher values of α result in a Rényi divergence whose terms with the greatest ratios between P and Q dominate the divergence, whereas as α approaches zero, dissimilarities are given less weight and the divergence becomes a more uniform measure over all outcomes.

Bhattacharyya Distance [7] measures the similarity between probability distributions. Given distributions P and Q, the Bhattacharyya distance is defined as:

$$D_B(P, Q) = -ln(BC(P, Q)) \tag{4.7}$$

where BC is the Bhattacharyya coefficient, defined as:

$$BC(P, Q) = \sum_{x} \sqrt{P(x)Q(x)} \tag{4.8}$$

4.3 Optimal Transport Measures

Optimal Transport measures focus on quantifying how one distribution can be transformed into another while minimising a given cost. These measures provide a framework for comparing probability distributions by considering the minimal "work" required to transform

one distribution into another.

Wasserstein Distance [45] is a metric for comparing probability distributions. Let P and Q represent two probability distribution, then, the Wasserstein distance quantifies the cost or the minimum "work" needed to transform P into the Q, where "work" is defined as the amount of distribution mass moved times the distance it needs to be moved. Wasserstein-1 is defined as:

$$W_1(P,Q) = \inf_{\gamma \in \Pi(P,Q)} \int_{\mathbb{R}\times\mathbb{R}} |x-y| d\gamma(x,y), \tag{4.9}$$

where $\Pi(P,Q)$ denotes the set of all couplings of P and Q.

In this work, we also make use of the specific quadratic case of the general Wasserstein distance where $p = 2$, often referred to as the Wasserstein-2 distance. This distance considers the square of the Euclidean distance as the cost function, leading to smoother and more regular optimal transport plans. Small changes in the distributions result in smoother and more predictable changes in the distance metric. The Wasserstein-2 distance is defined as:

$$W_2^2(P,Q) = \inf_{\gamma \in \Pi(P,Q)} \int_{\mathbb{R}\times\mathbb{R}} \|x-y\|^2 d\gamma(x,y). \tag{4.10}$$

4.4 Statistical Moments

Moments-based measures provide statistical properties that help characterise and understand a function such as a probability distribution. In our work, we make use of discrete probability distributions as representations. As such, we will define the various moments-based measures here as moments of discrete distributions. The n-th moment about the origin (i.e., zero) of a probability distribution (the *expected value* of X^n, also known as a *raw moment*) describes the most basic properties of a distribution. The moments about its mean μ provide additional quantities about the shape of a distribution (known as *central moments*). Moments which are *standardised* are dimensionless quantities that represent the distribution independent of the scale on which the data is measured, allowing for comparison across different distributions.

The first raw moment, the *mean*, describes the centre of the distribution, that is, it provides the expected value of the random variable. For a discrete distribution, the mean is given by:

$$\mu = E(X) = \sum xP(x) \tag{4.11}$$

The second central moment, the *variance*, gives insight into the spread of the distri-

bution around the mean μ, reflecting how much the values of the random variable deviate from the mean:

$$\sigma_2 = E((X-\mu)^2) = \sum x^2 p(x) - \mu^2 \tag{4.12}$$

The third standardised moment, *skewness*, describes the asymmetry of the distribution. For example, a distribution that has a longer tail on the left will have a negative skewness and a distribution that has a longer tail on the right will have a positive skewness:

$$\tilde{\mu}_3 = \frac{\mu_3}{\sigma_3} = \frac{E((X-\mu)^2)}{E((X-\mu)^2)^{\frac{3}{2}}} \tag{4.13}$$

The fourth standardised moment, *kurtosis*, provides information about the heaviness, height, or *peakedness* of the tail of the distribution, indicating how much of the distribution's probability mass is in the tails compared to the centre. Leptokurtic distributions, where a distribution has a heavy tail, will have a high kurtosis. Conversely, platykurtic or light-tailed distributions will have a low kurtosis value:

$$\tilde{\mu}_4 = \frac{\mu_4}{\sigma_4} = \frac{E((X-\mu)^4)}{E((X-\mu)^2)^{\frac{4}{2}}} \tag{4.14}$$

In our work, we use the mean μ, the variance σ_2, the skewness $\tilde{\mu}_3$, and kurtosis $\tilde{\mu}_3$ over single distributions to capture a comprehensive picture of the distribution, beginning from its basic central location and dispersion to more complex characteristics of its form.

While these moments provide more granular quantitative descriptions of our distribution-based representations, each successive moment only provides a finer level of detail about the the extremity of the tailedness of the distribution and as such, any higher moments are of little practical value to our work.

Alongside moments of the representations of single domains, we also make use of functions which match higher order moments or "divergence in a projected space" [46]. We make use of three *higher-order measures*.

Central Moment Discrepancy (CMD) [117] measures the distance between two distributions based on the k-th central moment of their distributions. Zellinger et al. [117] define CMD as follows: Let $X = \{X_1, X_2, ..., X_N\}$ and $Y = \{Y_1, Y_2, ..., Y_N\}$ be bounded random vectors independent and identically distributed from two probability distributions P and Q on the compact interval $[a, b]^N$. The central moment discrepancy metric (CMD) is defined by:

$$CMD(P,Q) = \frac{1}{|b-a|} \left\| E(X) - E(Y) \right\|_2 + \sum_{k=2}^{\infty} \frac{1}{|b-a|^k} \left\| c_k(X) - c_k(Y) \right\|_2 \tag{4.15}$$

where $E(X)$ is the expectation of X and

$$c_k(X) = E\left(\prod_{i=1}^{N}(X_i - E(X_i))^{r^i}\right) \tag{4.16}$$

is the central moment vector of order k where $r_1 + r_2 + r_N = k$ and $r_1...r_N \geq 0$. Similarly to the authors, we set $k = 5$ for computational efficiency.

Maximum Mean Discrepancy (MMD) [36] is a statistical distance measure used to compare distributions. MMD is a method used to measure how different two sets of data are. This is done by comparing the average values (means) of their characteristics. MMD maps samples to a mathematical space called Reproducing Kernel Hilbert Space (RKHS) to make this comparison more effective. In this space, MMD is able to calculate more accurately the differences between the averages of the features of the two data sets. Mathematically, given two random variables $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_n\}$ that are drawn from distributions P and Q and a feature mapping $\phi : \mathcal{X} \to \mathcal{H}$, the MMD is defined as:

$$MMD(X, Y) = \left\| \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(y_i) \right\|_{\mathcal{H}} \tag{4.17}$$

If distributions, P and Q, are identical, their average values (means) will also be the same. But if they're different, their averages will also be different, i.e. their MMD > 0. To accurately measure how different the two distributions are, especially when the differences are complex, we use a technique that involves representing these distributions in a space with more dimensions. In this higher-dimensional space, we apply special calculations called kernel functions. These functions help us understand and quantify even the subtle or complex differences between P and Q. In this work, we make use of the GeomLoss[1] package which implements MMD and different kernel functions. We use the default value of 0.05 for σ for each of the below kernels, respectively:

Energy Kernel

$$\phi(x, y) = -\left\|x - y\right\|_2 \tag{4.18}$$

Gaussian Kernel

$$\phi(x, y) = exp\left(-\frac{\left\|x - y\right\|_2^2}{2\sigma^2}\right) \tag{4.19}$$

Laplacian Kernel

$$\phi(x, y) = exp\left(-\frac{\left\|x - y\right\|_2}{\sigma}\right) \tag{4.20}$$

[1] https://www.kernel-operations.io/geomloss/

Correlation Alignment (CORAL) [103]: In domain adaptation, correlation alignment is used to align the second-order statistics of two distributions to minimise the discrepancy between domains. In particular, if d is the dimension of the representation, $\|\|\|_F$ is the Frobenius norm and Cov_I, Cov_T is the covariance matrix of the intermediate and target samples, CORAL minimises the Frobenius norm of the difference between intermediate and target covariance matrices:

$$D_{CORAL} = \frac{1}{4d^2} \|Cov_I - Cov_T\|_F^2 \tag{4.21}$$

While we do not align intermediate and target domains in this work, we use CORAL as a measure of the divergence between the second-order statistics of intermediate and target distributions.

4.5 Diversity Measures

Entropy [96]: For clarity, we redefine our definition of Entropy from Section 2.2.3. *Entropy*, $H(X)$ for a random variable X, is a measure of the uncertainty of randomness in the information context of X. Higher entropy implies greater unpredictability, whereas lower entropy indicates more predictability. Mathematically, for a discrete random variable with numerous possible outcomes $x_1, x_2, \ldots, x_n$ and respective probabilities $P(x_1), P(x_2), \ldots, P(x_n)$, then entropy is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i). \tag{4.22}$$

Rényi Entropy [87] extends the concept of entropy by offering a more generalised approach. It applies to a discrete random variable X with n possible values, each having a probability of p_i. Rényi entropy, is then defined as:

$$H_\alpha(X) = \frac{1}{\alpha - 1} log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right) \tag{4.23}$$

for $0 \leq \alpha \leq \inf$.

Simpson's Index [97]: Typically used in ecology to measure biodiversity, the value of the Simpson's index reflects diversity differences in populations and how evenly distributed the population is, it is defined as:

$$D = 1 - \sum_{i} \left(p_i^2 \right) \tag{4.24}$$

Chapter 5

Experimental Setup

This chapter outlines the methodology underpinning our transfer learning experiments, focusing on the transformation and analysis of datasets. The chapter is structured to provide a thorough overview of the datasets, including the reasons for their selection and the specifics of their modification. Additionally, it discusses the strategies employed for neural network model training and the methods used for computing divergence measures, all of which are central to our research.

In Section 5.1, we detail the datasets chosen for this study. The selection was based on their suitability for assessing transfer learning across different domains and tasks. This section explains the process of adapting these datasets to meet the specific requirements of our experiments. We discuss the transformation of dataset tasks, the establishment of uniform sample sizes across different experimental conditions, and the implementation of three distinct target training sample settings—zero-shot, few-shot, and limited. These modifications are essential for ensuring that our experiments accurately reflect real-world scenarios and for minimising potential confounding factors in our analysis.

In Section 5.2, we discuss our approach to model training for transfer learning. We discuss the specifics of training bottleneck adapter modules, including model optimisations and other considerations of our training strategy. This section provides an understanding of how the models are prepared and optimised for the transfer learning tasks, providing insight into the technical aspects that contribute to model performance.

5.1 Dataset Preparation and Transformation Strategies

In this section, we describe the datasets selected for this study and the rationale behind their selection. The selection of datasets is informed by their established use in related work and the distinct characteristics that make them suitable for analysing domain-specific aspects of transfer learning.

The Amazon Product Reviews [116] dataset was selected for its comprehensive cov-

erage of a wide array of products, providing a diverse and rich linguistic dataset. Each review typically includes a textual review and a star rating. The text can provide insights into the customer's opinion about the product, while the star rating is a direct indicator of sentiment (positive, negative, or neutral). The dataset covers a wide range of products, each with a substantial number of samples. More importantly, the dataset contains metadata distinguishing products by their domains, allowing for straightforward transformation to a topic classification task.

Multi-Domain Yelp Business Reviews [73], derived from the 2015 Yelp Dataset Challenge, was chosen due to its similarity to the Amazon dataset in terms of review structure but differs significantly in context, focusing on businesses and services. These reviews are written by users of the website, labelled ordinally by their star rating. Similar to Amazon reviews, the textual content and ratings can be used for sentiment analysis, however, in this work, we use the categories of businesses for topic classification.

In contrast to the product and service review datasets, the Yahoo! Answers [116] dataset offers a distinct format of user-generated content, comprising questions and answers across a wide spectrum of topics. This diversity introduces a different challenge, moving from opinion-based text to information-seeking and knowledge-sharing interactions. The dataset's inclusion allows us to test the adaptability of transfer learning models to a broader range of linguistic structures and content types, extending beyond reviews into more varied forms of user-generated content.

The Text REtrieval Conference (TREC) Incident Streams [59] dataset, designed for the TREC Incident Streams track, focuses on identifying actionable information from social media streams during emergency events. It contains short spans of text from social media posts and updates related to incidents such as natural disasters, accidents, or public safety events. The primary use of this dataset is in information retrieval and classification to quickly and accurately identify relevant information that can be used by emergency responders or for public information. The nature of the data tests the ability of models to process and interpret short, often fragmented and urgent messages, particularly important for real-world applications like emergency response.

5.1.1 Dataset Transformation Approach

Table 5.1 provides an overview of the datasets used in our transfer learning study, highlighting their original and transformed configurations. It includes the number of domains before ($\mathcal{D}_O$) and after transformation ($\mathcal{D}_{XFORM}$), the count of domain combinations (# Pairs), and sample sizes for both intermediate ($S_{\mathcal{D}_I}$) and target domains ($S_{\mathcal{D}_{T_{Train}}}$ and $S_{\mathcal{D}_{T_{Test}}}$ for the training and test sets, respectively) under various training conditions. This overview is essential for understanding the datasets' breadth and their application in our experiments.

Table 5.1: Overview of the selected datasets. This table summarises the key characteristics of each dataset, including the name, the abbreviation used in later chapters (Abbr.), the original number of domains ($\mathcal{D}_O$), the number of domains after filtering low-sample domains ($\mathcal{D}_{XFORM}$), and the number of domain combinations used (# Pairs), the number of samples for the intermediate domain's dataset (S_{D_I}), the number of samples for the target domain's training set for different experimental settings ($S_{\mathcal{D}_{T_{Train}}}$), and the number of samples for the target domain's test set ($S_{\mathcal{D}_{T_{Test}}}$).

Name	Abbr.	$\mathcal{D}_O$	$\mathcal{D}_{XFORM}$	# Pairs	$S_{\mathcal{D}_I}$	$S_{\mathcal{D}_{T_{train}}}$	$S_{\mathcal{D}_{T_{test}}}$
Amazon	AM	46	42	1722	25,000	50/1000	2500
Yelp	YL	22	16	240	25,000	50/1000	2500
Yahoo	YH	10	10	90	25,000	50/1000	2500
TREC-IS	IS	25	17	272	1,000	50	2500

Our methodology centres on the *One-vs-Rest* approach to divide larger datasets into smaller, binary classification tasks. This involves designating one domain as the positive class and combining all other domains and designating them as the negative class. For the Amazon and Yelp datasets, originally designed for rating prediction, we leveraged the metadata within each dataset to create distinct topic classification datasets based on their domains. For multiclass and multilabel topic classification datasets, we split these into binary equivalents using the same *One-Vs-Rest* strategy. Across all datasets, we set a uniform number of samples for the intermediate domain, $S_{\mathcal{D}_I}$, to remove sample size as a confounding variable in task selection.

To accurately replicate transfer learning scenarios, we implemented three distinct target training sample settings in our experiments, each reflecting varying data availability conditions typical in real-world applications. These settings are *Zero-shot* (ZS), *Few-shot* (FS), and *Limited* (LTD).

The zero-shot transfer scenario reflects a real-world situation where a user has no training data available for the target domain. This setting is important for assessing the inherent transferability between domains, as it tests their ability to apply knowledge gained from the intermediate domain to completely new domains without any domain-specific fine-tuning. It provides baseline insights into the fundamental generalisation capabilities of the models.

The few-shot scenario is designed to simulate conditions where very limited training data is available in the target domain. To determine the amount of samples to use in the few-shot setting, we follow the 50-sample few-shot upper bound set by Poth et al. [72] By restricting the target training sample size to 50, this setting provides a test of the models' ability to adapt to new domains with very little available data, as is often the case in niche research topics and other academic settings.

In the limited transfer setup, with a target training sample size of 1,000, we replicate

a scenario where there is an adequate but limited amount of training data in the target domain. In scenarios where there is a sufficient number of samples to learn from, there is a significant shift in the semantics of the data—known as *concept drift*—when the model is applied to a new domain, with completely new term usage patterns and document contexts. In this setting, we follow prior work in low-resource simulation by Phang et al. [77] and set the number of target domain training samples to 1,000.

The TREC-IS dataset, due to its limited sample size for both training and testing in numerous domains, necessitated an adjustment in the number of intermediate samples, fixed at 1,000. Consequently, this dataset is excluded from experiments in the *Limited* setting.

5.2 Model Training Strategies

5.2.1 Domain Adapter Generation

In our experimental setup, we use adapter-based training for multiple reasons, primarily driven by the scale of our experiments and the need to mitigate catastrophic forgetting. As seen in Table 5.1, the number of task pairs for each dataset is considerable, with Amazon (1722), Yelp (240), Yahoo (90), and TREC-IS (272). Excluding TREC-IS from the Limited setting, this amounts to a total of 6,700 intermediate-to-target model training tasks.

The use of a standard BERT-base model as a baseline in these scenarios presents a significant challenge in terms of memory requirements. A single BERT-base model occupies approximately 450 MB of memory. Training and storing models for all task pairs would demand around 3.05TB of storage space. Adapters offer a more efficient alternative. By training and saving only a few layers of the network, each adapter requires about 3.5MB of storage, reducing the total memory footprint to just 23.7GB for all 6,785 models. This makes adapter-based training a practical and resource-efficient approach for handling the large scale of our experiments.

Moreover, our training process involves first training models on intermediate tasks using BERT-base, and then retraining them for target tasks in the few-shot and limited settings. This sequential training increases the risk of catastrophic forgetting, a phenomenon where a neural network, when trained on a new task, tends to forget the knowledge it acquired from previous tasks. Catastrophic forgetting is particularly problematic in transfer learning scenarios, where a model is expected to retain and apply knowledge across various domains and tasks.

Adapters address this issue effectively. They update significantly fewer parameters in the model compared to full model training, which helps in retaining the knowledge previously acquired while still adapting effectively to new tasks. This selective parameter

updating is important for preserving the model's performance across a diverse range of tasks without the need for storing numerous large-scale models. In essence, adapter-based training strikes a balance between efficiency, memory usage, and the ability to mitigate catastrophic forgetting, making it a suitable choice for our extensive transfer learning experiments.

In our experimental design, the emphasis is not on extensive model tuning but rather on developing a generalised approach for assessing transferability. To this end, we utilise standard bottleneck adapters equipped with a classification head, employing the default configuration (`PfeifferConfig` [70]) as provided by the AdapterHub [69] package. This approach ensures that our methodology remains broadly applicable and not overly specialised to specific tuning settings. We adhere to the recommended hyperparameters from the AdapterHub authors, setting our learning rate at $1e-4$ for consistency and reliability across all models.

The training process for our models is conducted sequentially. For each intermediate-to-target model, we begin by loading the adapter module trained on the corresponding intermediate task. We then activate this adapter for further training, fine-tuning it on the target task. After this process, we save a new version of the adapter. This sequential training and fine-tuning approach allows each model to build upon the knowledge acquired in the intermediate task, adapting it effectively to the new target task while maintaining a streamlined and consistent training procedure across all models.

5.2.2 Optimising Models for Efficient Training

Our model training process incorporates Mixed Precision Training, as outlined in Section 2.4.3, to enhance the end-to-end efficiency of our framework. This technique uses both half-precision and single-precision floating-point formats, enhancing training efficiency by reducing computational demand while preserving model accuracy.

In practical terms, we implement mixed precision by dynamically adjusting computational precision during the training process. This approach optimises training speed and reduces memory consumption. Alongside this, we employ automated gradient scaling, which is essential for maintaining the precision and range of gradients when training in lower precision formats. Gradient scaling counteracts the limitations of half-precision computation, ensuring that small gradient values do not vanish due to reduced precision. To further stabilise our training, we incorporate gradient clipping. This technique prevents the issue of exploding gradients by imposing a threshold on the magnitude of the gradients during backpropagation. By doing so, we maintain the stability and consistency of the training process.

5.2.3 Cost Estimation

An integral part of our experimental framework is the tracking of resource consumption, specifically focusing on CO_2 emissions and total kilowatt-hours (kWh) incurred. For this purpose, we use the CodeCarbon [55] package, a tool designed to estimate the carbon footprint and energy usage of machine learning models. Understanding the environmental impact of building representations and training models is particularly important as the field of machine learning advances and the computational demands of training larger models increase. Recording these metrics allows us to estimate the environmental benefits of our transferability estimation framework.

We systematically record emissions for each logical grouping of tasks in our experiments. For instance, when building a vocabulary of term frequencies, we log a single emissions task. In the case of model training and inference, we record emissions for both processes separately. This detailed tracking enables us to gain insights into the environmental cost at different stages of our experimental pipeline.

Hardware Setup: Our experiments are conducted on a hardware setup comprising three NVIDIA® TITAN™ RTX GPUs, each offering 130 Tensor TFLOPs of performance, 576 tensor cores, and 24 GB of GDDR6 memory. The computational tasks are managed by an Intel® Xeon® Gold 5222 Processor (16.5MB Cache, 3.80 GHz) with 96GB of RAM, distributed across three docker containers. We maintain the same hardware configuration across all of our experiments ensuring that we reliably report the resource consumption and environmental footprint of our experiments.

5.3 Domain Transfer Analysis

In this section, we turn our attention to understanding how well transfer learning works in challenging scenarios. Specifically, we focus on the few-shot setting, where models have very limited training data. Here, we present tables showing the performance of different tasks when we apply transfer learning strategies.

Table 5.2 lists how intermediate tasks (rows) impact the performance on target tasks (columns). For the sake of legibility, we enumerate the tasks (e.g. $\mathcal{T}_1, \mathcal{T}_1, \ldots, \mathcal{T}_n$). The table uses colour coding to show performance changes: green for gains and orange for losses. The intensity of the colour indicates how big the gain or loss is. Values in the table represent F_1-score performance, comparing to the baseline performance scores (denoted *No Transfer*) when only trained with target task training data. Additionally, *Avg. Transfer* shows the average of the transferred ($\mathcal{D}_I \rightarrow \mathcal{D}_T$) scores.

Table 5.2 shows that choosing the right tasks for transfer learning (highlighted in bold) leads to significant improvements compared to the baseline. The biggest increase in

Table 5.2: Target task performance scores when transferring from intermediate tasks (rows) to target tasks (columns) for the Yahoo dataset in the *Few-shot* setting. Tasks are enumerated and prepended with $\mathcal{T}$ for legibility. Values in bold denote the highest transfer performance. *No Transfer* denotes the baseline performance of models when training only on target task training data. *Avg. Transfer* denotes the average of $\mathcal{D}_I \rightarrow \mathcal{D}_T$ transfer performance.

Task ($\mathcal{T}$)	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$	$\mathcal{T}_9$	$\mathcal{T}_{10}$
No Transfer	0.459	0.837	0.607	0.638	0.702	0.763	0.689	0.758	0.648	0.823
Avg. Transfer	0.646	0.882	0.652	0.744	0.840	0.853	0.802	0.796	0.705	0.875
$\mathcal{T}_1$	-	0.905	0.716	0.749	0.857	0.853	0.820	0.741	0.737	0.872
$\mathcal{T}_2$	0.697	-	0.654	**0.814**	0.852	0.874	0.696	**0.860**	0.681	0.833
$\mathcal{T}_3$	0.695	0.864	-	0.675	0.830	0.825	**0.857**	0.845	0.693	0.896
$\mathcal{T}_4$	0.621	0.852	0.680	-	0.832	0.772	0.784	0.808	0.744	0.843
$\mathcal{T}_5$	0.637	0.845	0.647	0.756	-	**0.891**	0.780	0.782	**0.774**	0.883
$\mathcal{T}_6$	0.660	0.918	0.675	0.720	0.813	-	0.818	0.854	0.641	**0.901**
$\mathcal{T}_7$	**0.739**	0.868	0.637	0.737	0.818	0.867	-	0.770	0.728	0.882
$\mathcal{T}_8$	0.682	**0.923**	**0.751**	0.755	0.857	0.840	0.814	-	0.664	0.887
$\mathcal{T}_9$	0.591	0.851	0.615	0.742	0.828	0.886	0.819	0.754	-	0.876
$\mathcal{T}_{10}$	0.495	0.912	0.494	0.748	**0.871**	0.864	0.831	0.755	0.679	-

performance is 61% when transferring from task $\mathcal{T}_7$ to $\mathcal{T}_1$. The smallest increase from the best task combination is a 9.5% gain, seen when transferring from $\mathcal{T}_6$ to $\mathcal{T}_{10}$. These results clearly show that transfer learning can greatly improve performance, especially when we have limited data to train on.

On the other hand, using the least effective task combinations can lead to much smaller gains or even decrease performance. For instance, if we consider the same target domains, but choose the least effective task, transferring from $\mathcal{T}_{10}$ to $\mathcal{T}_1$ results in only a 7.8% improvement, and transferring from $\mathcal{T}_2$ to $\mathcal{T}_{10}$, results in a minimal 1.2% gain. The consequences of a poor choice are more stark in some cases, like when transferring from $\mathcal{T}_{10}$ to $\mathcal{T}_3$, which leads to an 18.6% drop in performance. This is in stark contrast to the 23.72% improvement seen with the optimal task combination of $\mathcal{T}_8$ to $\mathcal{T}_3$. These results demonstrate the importance of accurately predicting the most effective task combinations for transfer learning beforehand. The stark contrast between the highest gains and the losses incurred with less effective combinations underscores the potential impact of informed task selection. Given these insights, we now move to analyse the TREC-IS dataset, focusing on crisis classification. This area presents even more challenges, and the consequences of choosing the wrong task for transfer learning are more pronounced.

Table 5.3 shows how transfer learning affects performance on the TREC-IS dataset's target tasks. Similar to our previous table, tasks are labelled as $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$. The table uses colours to indicate performance gains and losses, with the colour intensity reflecting the magnitude.

This table reveals a key insight: In the TREC-IS dataset, it's more common to see a significant decrease in performance than a significant increase when applying transfer

Table 5.3: Target task performance scores when transferring from intermediate tasks (rows) to target tasks (columns) for the TREC-IS dataset in the *Few-shot* setting. Tasks are enumerated and prepended with $\mathcal{T}$ for legibility. Values in bold denote the highest transfer performance. *No Transfer* denotes the baseline performance of models when training only on target task training data. *Avg. Transfer* denotes the average of $\mathcal{D}_I \rightarrow \mathcal{D}_T$ transfer performance.

Task ($\mathcal{T}$)	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$	$\mathcal{T}_6$	$\mathcal{T}_7$	$\mathcal{T}_8$	$\mathcal{T}_9$	$\mathcal{T}_{10}$	$\mathcal{T}_{11}$	$\mathcal{T}_{12}$	$\mathcal{T}_{13}$	$\mathcal{T}_{14}$	$\mathcal{T}_{15}$	$\mathcal{T}_{16}$	$\mathcal{T}_{17}$
No Transfer	0.623	0.607	0.627	0.595	0.651	0.655	0.756	0.549	0.633	0.636	0.609	0.602	0.572	0.763	0.652	0.596	0.689
Avg. Transfer	0.595	0.609	0.583	0.589	0.626	0.592	0.648	0.562	0.610	0.595	0.583	0.581	0.547	0.668	0.641	0.549	0.678
$\mathcal{T}_1$	-	0.629	0.647	0.599	0.659	0.633	0.590	0.521	0.647	0.504	0.529	0.611	0.502	0.711	0.707	0.529	0.704
$\mathcal{T}_2$	0.579	-	0.611	0.653	0.596	0.633	0.644	0.587	0.639	0.594	0.535	0.557	**0.634**	0.613	0.710	0.514	**0.802**
$\mathcal{T}_3$	0.674	0.637	-	0.533	0.470	0.652	0.650	0.449	0.649	0.654	**0.659**	0.538	0.616	0.673	0.588	0.487	0.683
$\mathcal{T}_4$	0.605	0.629	0.634	-	0.638	0.638	0.675	0.580	0.593	0.427	0.562	0.630	0.560	0.599	0.665	0.320	0.780
$\mathcal{T}_5$	0.611	0.658	0.540	0.636	-	0.587	0.576	0.636	0.623	0.644	0.627	0.620	0.566	0.632	0.704	0.561	0.672
$\mathcal{T}_6$	0.673	0.526	0.654	0.552	0.641	-	0.748	0.623	0.632	0.662	0.557	0.563	0.577	0.719	0.546	0.587	0.627
$\mathcal{T}_7$	0.677	0.438	0.605	0.463	0.606	0.655	-	0.376	0.556	0.610	0.529	0.636	0.529	0.688	0.648	0.596	0.429
$\mathcal{T}_8$	0.312	0.636	0.592	0.577	0.670	0.545	0.476	-	0.633	0.618	0.643	0.505	0.548	0.624	0.639	0.631	0.700
$\mathcal{T}_9$	0.488	0.629	0.578	0.601	0.650	0.567	0.706	0.535	-	**0.680**	0.494	0.577	0.552	0.712	0.567	0.617	0.580
$\mathcal{T}_{10}$	0.661	0.666	0.508	0.619	**0.697**	0.553	0.755	**0.658**	**0.669**	-	0.598	**0.650**	0.558	0.727	0.660	**0.622**	0.669
$\mathcal{T}_{11}$	0.571	0.431	0.544	0.657	0.659	0.538	0.716	0.633	0.613	0.634	-	0.632	0.600	0.678	0.689	0.576	0.752
$\mathcal{T}_{12}$	0.622	0.630	0.391	0.533	0.641	0.514	0.618	0.580	0.543	0.547	0.622	-	0.520	0.754	**0.713**	0.563	0.673
$\mathcal{T}_{13}$	0.562	0.675	0.598	**0.679**	0.633	0.552	0.692	0.614	0.626	0.607	0.600	0.581	-	0.674	0.524	0.554	0.739
$\mathcal{T}_{14}$	0.690	0.615	**0.688**	0.637	0.651	0.598	0.646	0.644	0.459	0.664	**0.659**	0.623	0.432	-	0.642	0.484	0.712
$\mathcal{T}_{15}$	**0.716**	**0.690**	0.567	0.576	0.578	0.626	0.720	0.479	0.650	0.595	0.530	0.554	0.475	0.649	-	0.580	0.704
$\mathcal{T}_{16}$	0.502	0.625	0.576	0.449	0.592	0.577	0.727	0.540	0.608	0.622	0.595	0.441	0.563	0.734	0.641	-	0.628
$\mathcal{T}_{17}$	0.576	0.636	0.592	0.656	0.640	0.612	0.433	0.538	0.618	0.461	0.592	0.574	0.524	0.497	0.609	0.562	-

learning. Indeed, transfer learning leads to improvements over the baseline in only 33% of cases (80 out of 242), a marked difference from the 94.4% improvement rate observed with the Yahoo dataset. This trend is further highlighted when considering the average transfer scores for each target domain, which are mostly lower than the baseline performances, except for two domains.

To emphasise the risks of choosing the wrong tasks, consider tasks $\mathcal{T}_1$ and $\mathcal{T}_8$. When transferring from the best-suited intermediate tasks, $\mathcal{T}_{15}$ and $\mathcal{T}_{10}$, we see performance gains of 14.9% and 19.9%, respectively. However, if we randomly end up choosing the least effective tasks, $\mathcal{T}_8$ and $\mathcal{T}_3$, the performance drops dramatically by 49.8% and 12.8%.

The occurrence of negative transfer in the TREC-IS dataset can be attributed to several factors. First, the tasks in this dataset may have more diverse data distributions compared to those in the Yahoo dataset. When the intermediate and target tasks have significantly different data distributions, the transferred knowledge may not be directly applicable or may even interfere with learning on/predicting the target task. Second, the complexity of the tasks in the TREC-IS dataset might vary more than those in the Yahoo dataset. Transferring from a simpler task to a more complex one, or vice versa, can lead to suboptimal performance as the model struggles to adapt to the new task's intricacies. Finally, the pretraining phase on the intermediate tasks may have learned patterns or features that are not relevant or even contradictory to the target task, leading to negative transfer.

To mitigate the risks of negative transfer, it is important to carefully select the intermediate tasks that are most likely to benefit the target task. This selection process should consider factors such as task similarity, data distribution compatibility, and the transferability of learned features. By developing a framework that can accurately predict the

most beneficial task combinations, we can minimise the occurrences of negative transfer and optimise the performance gains achieved through transfer learning.

In summary, our analysis across both the Yahoo and TREC-IS datasets in a few-shot setting presents a clear picture: the choice of tasks in transfer learning can lead to significant performance variations. While the right task combinations can yield substantial improvements, as seen with gains up to 61% in Yahoo, the wrong choices can result in minimal benefits or substantial drops in performance, such as the 49.8% decrease observed in the TREC-IS dataset. These results highlight the motivation for an effective framework that can accurately predict the most beneficial task combinations for transfer learning. Such a framework is particularly essential when the cost of poor performance carries significantly more weight, such as that of models designed for crisis classification tasks. By enabling more precise and informed task choices, we can maximise the potential of transfer learning.

Building on these insights, we now transition to our first experimental chapter, where we focus on evaluating distributional representations. Our goal is to understand which characteristics of distributional representations are most beneficial in identifying the most effective task combinations.

Chapter 6

Distributional Representations

6.1 Introduction

This chapter focuses on distributional representations within the field of natural language processing (NLP) for estimating transferability. Distributional representations, which primarily focus on term frequency and linguistic patterns, form the baseline for understanding text in computational models. The idea of a distributional *semantic space* has its origins in structural linguistics and information theory. Zellig S. Harris' work [38] on distributionalism laid foundational principles for viewing language through a mathematical lens and by extension, modern computational linguistics and natural language processing. The *distributional hypothesis* introduced by Harris suggested that words or lexemes that are used and occur in similar contexts tend to have similar meanings. The transition from foundational theories to practical models in NLP finds a significant milestone in the conceptualisation of the *vector space model* (VSM) by Salton et al. [93], effectively operationalising the principles of the distributional hypothesis. In this framework, words are transformed into vectors within a multi-dimensional semantic space, quantifying the idea that "a word is characterized by the company it keeps" [31]. This mathematical representation enables the encoding of term patterns and relationships in a form amenable to computational processing. VSMs essentially materialise the abstract principles of distributional semantics into a concrete, computational model, bridging the gap between theoretical linguistics and practical NLP applications.

"The DH [distributional hypothesis] states that the semantic similarity of lexical items is a function of their distribution in linguistic contexts" [51]. Hence, by comparing the distributional patterns of words and phrases across different corpora, one can gauge the degree of semantic overlap between them. This comparative approach plays a particularly important role in transfer learning, where the task is to leverage knowledge acquired from one domain to improve performance on another. The success of transfer learning rests on the supposition that successful knowledge transfer relies on the existence of linguistic

overlap between the domains involved. Recent work in domain divergence empirically supports this notion. Ben-David et al. [3] found that the performance on a target domain is largely bounded by its divergence to the *source*—or in our case, *intermediate*—domain. Hence, approximating the divergence between distributional representations can serve as a useful method for assessing the potential of successful transfer.

In this work, we leverage statistical measures (introduced and defined in Chapter 4) to quantify the divergence between pairs of distributional representations derived from domain-specific corpora. Our approach involves a novel, comprehensive aggregation analysis and an analysis of vocabulary construction methods used for constructing term distributions, measuring representation quality through Spearman's ρ correlation between domain divergence and transfer learning performance. Through accurately estimating the success of transfer between domains, we significantly streamline the process of transfer learning, efficiently circumventing the extensive and often unproductive search for effective task combinations.

This chapter aims to evaluate the methods used to construct distributional representations. Our analysis is centred on three main categories, each selected for their proven [46, 68, 91] effectiveness in capturing domain divergence: Term Distributions (TD), Linguistic Feature Distributions (LFD), and Topic Frequency Distributions (KFD).

In the section on Term Distributions (TD), we examine the frequency analysis of terms using Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) approaches. This involves experimenting with various vocabulary construction techniques to establish shared vocabularies. We also explore divergence computation at different levels of abstraction, such as centroid and domain, to understand the variations in term usage across different domains.

Next, the Linguistic Feature Distributions (LFD) section delves into the analysis of categorical frequencies of linguistic features. This includes a thorough examination of features such as Universal POS (UPOS) tags, eXtended POS (XPOS) tags, Named Entity Recognition (NER), and Linguistic Dependency (DEP) frequencies. Our approach measures divergence across multiple levels, shedding light on the syntactic and semantic structures within various text corpora.

Finally, in the Topic Frequency Distributions (KFD) section, we use a BERTopic model with a k-means backend to construct topic distributions. This part of the chapter is aimed at gaining insights into the thematic elements distributed across different domains and how they contribute to our understanding of domain divergence.

Our approach to analysing these distributional representations involves a systematic evaluation of how well these representations can predict the success of transfer learning by correlating them with intermediate-to-target model performance scores. In line with this objective, the structure of the chapter is as follows:

1. In Section 6.2, we detail the methods used for constructing each type of distributional representation, focusing on the procedural aspects and the rationale behind each method.

2. Section 6.3 outlines specific research questions that guide the investigation of distributional representations, linking them back to the overarching goals of the thesis.

3. Sections 6.4, 6.5, and 6.6 include the results of applying statistical divergence measures to the distributional representations. The focus will be on how well these representations correlate with task suitability for transfer learning.

4. In Section 6.8, we conclude with a summary of findings, discussing the implications of these results for understanding transfer learning in NLP and setting the stage for subsequent analyses in the thesis.

6.2 Methodology

In this section, we present a systematic approach to evaluating distributional representations in the context of transfer learning for language tasks. Our focus is on three distinct types of representations: Term Distributions (TD), Linguistic Feature Distributions (LFD), and Topic Frequency Distributions (KFD). Each representation type is assessed for its capacity to capture linguistic features that describe the underlying domain, such that we can readily select similar domains for transfer.

The primary aim of this study is to evaluate the efficacy of these distributional representations in predicting the success of transfer learning, especially in scenarios where data resources are inherently limited. Such scenarios are not just theoretical constructs but are reflective of common challenges in real-world applications. They include Zero-shot (ZS), Few-shot (FS), and Limited (LTD) settings, which are frequently encountered in numerous fields where data scarcity is a prevalent issue. This section details the construction and analysis process for each representation type, highlighting technical steps and challenges encountered.

6.2.1 Spearman's Correlation Analysis

In this section, we discuss how Spearman's correlation analysis (Section 2.2.5) is used in this chapter to gauge the relationship between the divergence of domain representations and the relative transfer gain among intermediate-to-target models. Our aim is to discern whether a statistically significant inverse relationship exists, postulating that enhanced performance correlates with reduced divergence between domains.

Spearman's Correlation in Evaluating Transfer Gain. Building on the foundational understanding of Spearman's rank correlation coefficient (Spearman's ρ) established in the Section 2.2.5, this chapter uses Spearman's correlation as a tool for analysing the dynamics between domain divergence and transfer learning performance. Our hypothesis posits an inverse relationship, suggesting that as domain divergence decreases, the effectiveness, as quantified by F_1-score performance of intermediate-to-target models, of transfer learning increases.

We focus primarily on the strength of the correlation: a higher magnitude (closer to 1 or -1) implies a stronger predictive relationship between domain similarity and transfer learning efficacy. This approach enables us to quantify and evaluate the "quality" of our domain representations in terms of their predictive power for transfer learning success. In doing so, we develop an understanding of which domain representations and aggregation methods are most conducive to successful transfer learning, informing subsequent experimental work in intermediate task selection.

Divergence Computation. In this work, we use both distributional and embedding representations. For distributional representations at the document-level, we perform an L_1-normalisation over the frequency distributions to create a probability distribution of observations. When aggregating to a grouping of document frequencies, such as *centroid*- or *domain*-level aggregation—which we discuss in the following section—we sum over the dimension to aggregate and L_1-normalise the resultant frequencies. For embedding representations, we do not normalise the data unless performing an aggregation, where we use the mean over the desired aggregation dimension. The following section will expand on the aggregation strategies used in this work.

6.2.2 Aggregation Analysis

The initial phase of our evaluation process focuses on analysing baseline, domain-aggregated representations. This approach is essential both for computational efficiency and for gaining methodological insights into the characteristics of domains.

In the case of distributional representations, we consider corpus-wide frequencies of terms, linguistic categories, or topics, aggregating these frequencies in two primary ways: either over randomly selected subsets of documents to form a *centroid* (CTD) of frequencies, or over the entire corpus to create a *domain*-level (DOM) frequency distribution. Such aggregation is particularly important for distributional representations given the computational complexity and potential unreliability associated with sparse distributions, as the infrequent occurrence of specific terms or linguistic features across documents can lead to inaccurate assessments of their distributional characteristics.

In the case of embedding representations, we consider corpus-wide document vectors,

aggregating again to either centroids of document vectors or domain-level vectors. Our analysis involves several aggregation strategies:

Domain-to-Domain ($DOM - DOM$): This method involves aggregating data across the entire corpus for each domain, offering a comprehensive view of domain characteristics. This approach is beneficial for both high-dimensional distributional representations and embedding representations. By focusing on the aggregated level, we can capture overarching patterns and characteristics more effectively than at the individual document level, while also reducing computational demands.

Instance-to-Domain ($INST - DOM$): In this approach, individual instances (documents or vectors) from the intermediate domain ($\mathcal{D}_I$) are compared against the aggregated representation of the target domain ($\mathcal{D}_T$). This method is suitable for examining how individual instances relate to broader domain characteristics, applicable to both lower-dimensional and higher-dimensional representations.

Centroid-to-Domain ($CTD - DOM$): This strategy clusters subsets of data within the intermediate domain and compares these clusters (or *centroids*) against the aggregated representation of the target domain. This method is expected to yield insights particularly in cases where individual instances may not fully represent domain characteristics, due to sparsity or other factors.

Instance-to-Instance ($INST - INST$): This direct comparison between randomly selected instances from both domains provides a granular perspective. While offering detailed insights, this method might present challenges in consistency, especially in the context of sparse or high-dimensional data. It is most informative in scenarios where individual instances are sufficiently representative.

Centroid-to-Centroid ($CTD - CTD$): Comparing clusters or centroids between domains, this method aims to reveal inter-cluster relationships and thematic connections. It is particularly effective for understanding how larger groupings of data within domains relate to each other, offering a more stable comparison than instance-level analysis.

For centroid-to-domain, centroid-to-centroid, and instance-to-instance comparisons, we use parameters K, $K_{\mathcal{D}_I}$, and $K_{\mathcal{D}_T}$ (Appendix A.1), as determined by Bayesian hyperparameter optimisation. These parameters define the number of clusters and the cluster sizes for intermediate and target domains. To ensure comparability across different divergence approaches, these parameter values are kept consistent throughout the analysis.

6.2.3 Term Distributions

Term Distributions in this study are analysed using Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF). TF is a count of how often a term appears in a document, offering a basic but direct measure of term importance. Conversely, TF-IDF weighs a term's frequency against its prevalence across all documents, adjusting the weight assigned to terms that are unique to specific documents. These measures are effective for comparing linguistic patterns as they reflect both the frequency and specificity of term usage, providing insights into the unique linguistic characteristics of each domain.

Our initial, baseline performance analysis focuses on the divergence between domain-aggregated frequencies across each corpus. Aggregating frequencies over the corpus addresses the challenges of high-dimensional term distributions and sparse data at the document level. This aggregation is expected to yield more accurate approximations of divergence, capturing overarching linguistic patterns across a domain in a single, computationally efficient representation. In this stage, we make use of all of the aforementioned divergence measures to provide an overview of the baseline performance of these distributions.

The subsequent aggregation analysis uses the the best measure(s) from the previous experiment as a benchmark for comparison. This phase involves an examination of term distributions at varying levels of abstraction, specifically cluster-domain and cluster-cluster divergence approaches. We aim to explore how different levels of data abstraction influence the divergence measures, thus offering insights into the consistency and variability of linguistic patterns within and across domains.

Our final experiment involves varying approaches to vocabulary construction. The Term Ranking Method (TRM) determines term selection, either by TF or TF-IDF ranking to a specific vocabulary size. The Domain Context (DC) assesses the impact of using either target-specialized vocabularies or those constructed from a broader range of terms across all domains. To ensure a fair comparison, a fixed vocabulary size is set for each dataset, derived from the minimum number of unique terms across all target sets, removing vocabulary size as a confounding variable. The predefined vocabulary sizes are: 3000 for *Amazon* (AM), 6500 for *Yelp* (YL), 7000 for *Yahoo* (YH), and 3500 for *TREC-IS* (IS), each based on the domain with the fewest unique terms within the respective dataset.

6.2.4 Linguistic Feature Distributions

The Linguistic Feature Distributions in this study encompass Named Entity Recognition (NE), Linguistic Dependency (DEP), and Part-of-Speech (POS) tagging, specifically using Universal POS (UPOS) [64] labels and eXtended POS (XPOS) [94] label sets. These features are extracted using the EntityRecognizer, DependencyParser, and Tagger pipelines

from the spaCy Transformers' version of pretrained models. In particular, we count the presence of these tags for each document, and create a distribution over the associated label set. NE distributions focus on the occurrence and types of named entities within the text, DEP distributions analyse the patterns of linguistic dependencies, and U/XPOS distributions examine the usage of universal and extended POS tags. These categorical distributions provide rich insights into the syntactic and semantic structures of texts across different domains.

Similar to the approach with Term Distributions, we conduct an aggregation analysis for Linguistic Feature Distributions. However, given the categorical nature of these features, which results in less sparsity, we also include instance-domain and instance-to-instance comparisons in our analysis. This expansion allows us to capture both the broad domain-level linguistic patterns and the finer, more detailed structures observable in individual instances or documents. In instance-domain comparisons, individual text instances are compared against aggregated domain representations to assess how well individual documents align with or deviate from general domain characteristics. The instance-to-instance comparison, on the other hand, involves analysing linguistic feature distributions between individual documents, offering a direct perspective on the variability or similarity of linguistic structures at a more granular level.

6.2.5 Topic Frequency Distributions

The analysis of Topic Frequency Distributions begins with determining the optimal number of clusters, denoted as k. To achieve this, we employ the MiniBatchKMeans [95] algorithm, a computationally efficient variant of the k-means algorithm that uses mini-batches for processing, thereby reducing computation time without compromising the quality of the clustering outcome. The key objective at this stage is to identify the optimal k value for each target domain, which is accomplished through silhouette analysis. This method assesses the degree of separation between clusters, guiding us to select a k value that maximises the distinctiveness of each cluster within the target domain.

With the optimal k values identified, we proceed to train a BERTopic [37] model for each target domain. In this stage, all other domains within the same dataset serve as training samples. Although BERTopic typically employs HDBScan for clustering, we opted for the hard clustering method of k-means instead. This choice was motivated by several factors. First, k-means unambiguously assigns each document to one of the k clusters, which is important for accurately comparing the frequencies of particular categories that represent a specific domain. Hard clustering ensures that each document belongs to a single cluster, making the comparison of cluster distributions across domains more straightforward and interpretable. Second, the use of k-means simplifies the model training process and ensures more definitive cluster assignments. HDBScan's soft clustering

approach, while flexible, requires extensive hyperparameter tuning to achieve optimal results. Given the computational constraints and the need for a consistent and efficient framework, k-means provides a more practical solution.

Upon training the topic models, our focus shifts to evaluating the effectiveness of these models in predicting optimal task pairs for transfer learning. For each intermediate task, we transform the training documents into a distribution of topics using the model trained on the respective target domain. These topic distributions are then compared with the topic distribution of the target domain.

6.3 Research Questions

RQ1: What is the relative effectiveness of geometric, information-theoretic, and statistical moments-based divergence measures in capturing the correlation between Term Distributions (TD) and transfer learning performance, as indicated by Spearman's correlation with model performance scores?

This research question involves the comparative analysis of different divergence measures—geometric, information-theoretic, and statistical moments-based—in the context of Term Distributions. The goal is to determine which measure most accurately reflects the relationship between the characteristics of Term Distributions and transfer learning performance. By correlating these divergence measures with transfer learning model performance scores, this aims to identify the most relevant and effective measures.

RQ2: How do different aggregation methods (centroid-domain and centroid-to-centroid) influence the representational quality and correlation of Term Distributions with transfer learning performance compared to baseline domain-aggregated methods?

This question examines the influence of varying aggregation methods, specifically centroid-domain and centroid-to-centroid, on the effectiveness of Term Distributions in the context of transfer learning. The focus is to compare these methods against the baseline domain-aggregated approach to see how they affect the representational quality and their correlation with transfer learning performance.

RQ3: In constructing vocabularies for Term Distributions, how do different Term Ranking Methods (TRM) and Domain Contexts (DC) affect the representational efficacy and its correlation with transfer learning performance?

RQ3 involves assessing the impact different approaches to vocabulary construction for

Term Distributions have on the correlation strength. Specifically, it looks at the effects of varying Term Ranking Methods (TRM) and Domain Contexts (DC) on the representational effectiveness of Term Distributions and their correlation with relative transfer gain among intermediate-to-target model performance scores.

RQ4: Which divergence measures most effectively capture the correlation between Linguistic Feature Distributions (LFD) and transfer learning performance, as reflected in Spearman's correlation with model performance scores?

This question aims to identify the most effective divergence measures for Linguistic Feature Distributions (LFDs) by examining their correlation with transfer learning performance. By analysing the baseline performance of LFDs across different divergence measures and correlating them with transfer learning model performance scores, we seek to establish a foundational understanding of how different divergence measures interact with LFDs.

RQ5: How do different levels of aggregation (including instance-domain, instance-instance, centroid-domain, and centroid-centroid) influence the quality and performance correlation of Linguistic Feature Distributions in transferability estimation?

The focus of this research question is on the impact of different levels of aggregation, including instance-domain, centroid-domain, and centroid-centroid, on the effectiveness of Linguistic Feature Distributions in transfer learning tasks. This investigation seeks to understand how the choice of aggregation level affects the quality and performance correlation of LFDs.

RQ6: How do Topic Frequency Distributions, analysed at the domain-level, correlate with transfer learning performance across various divergence measures?

This research question explores the relationship between Topic Frequency Distributions, analysed at the domain-level, and their correlation with transfer learning performance. Through examining the performance of these distributions across various divergence measures, the goal is to understand how topic frequency distributions can be used effectively in transferability estimation.

6.4 Term Distributions Evaluation

The primary aims of our baseline experiments are two-fold. Our primary objective is to quantitatively evaluate the effectiveness of these representations, such that we can compare

Table 6.1: Spearman's ρ correlations between domain (DOM) representation divergence and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of metrics tested ($p < 0.006$). Bold values highlight the strongest correlations within each distribution group. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to their respective Cos. baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

	Zero-shot				Few-shot				Limited		
Measure	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
TF Distributions											
Cos.	-0.733*	-0.794*	-0.830*	-0.631*	**0.350***	0.084	**0.399***	0.024	**0.516***	0.293*	0.491*
L_1	-0.872*▲	-0.814*▲	-0.752*▼	**-0.781***▲	0.275*▼	-0.105	0.394*▼	0.013	0.490*▼	0.081	**0.497***▲
L_2	-0.681*▼	-0.778*▼	-0.667*▼	-0.379*▼	0.290*▼	0.099	0.321*▼	0.031	0.448*▼	**0.305***▲	0.353*▼
JS	**-0.876***▲	-0.848*▲	-0.798*▼	-0.769*▲	0.272*▼	-0.036	0.393*▼	-0.007	0.491*▼	0.152	0.489*▲
Rényi	-0.857*▲	-0.850*▲	**-0.863***▲	-0.763*▲	0.180*▼	-0.048	0.365*▼	0.012	0.384*▼	0.156*	0.439*▲
Bhat.	-0.875*▲	**-0.856***▲	-0.807*▲	-0.758*▲	0.268*▼	-0.020	0.397*▼	-0.010	0.489*▼	0.172*	0.496*▲
Wass-1	-0.681*▼	-0.778*▼	-0.667*▼	-0.379*▼	0.290*▼	0.099	0.321*▼	0.031	0.448*▼	0.304*▲	0.353*▼
Wass-2	-0.681*▼	-0.778*▼	-0.667*▼	-0.379*▼	0.290*▼	0.099	0.322*▼	0.031	0.448*▼	**0.305***▲	0.353*▼
TF-IDF Distributions											
Cos.	-0.848*	-0.848*	**-0.829***	**-0.769***	**0.260***	-0.012	**0.407***	0.032	**0.487***	**0.192***	**0.507***
L_1	**-0.858***▲	-0.824*▼	-0.736*▼	-0.744*▼	0.214*▼	-0.121	0.378*▼	-0.004	0.430*▼	0.057	0.485*▼
L_2	-0.773*▼	-0.847*▼	-0.742*▼	-0.706*▼	0.234*▼	-0.025	0.344*▼	0.029	0.450*▼	0.184*▼	0.418*▼
JS	-0.844*▼	-0.850*▲	-0.773*▼	-0.719*▼	0.204*▼	-0.044	0.384*▼	0.001	0.424*▼	0.139	0.481*▼
Rényi	-0.848*	-0.844*▼	-0.816*▼	-0.718*▼	0.190*▼	-0.044	0.376*▼	0.005	0.410*▼	0.153	0.455*▼
Bhat.	-0.835*▼	**-0.856***▲	-0.775*▼	-0.707*▼	0.199*▼	-0.025	0.384*▼	0.004	0.419*▼	0.160*▼	0.486*▼
Wass-1	-0.773*▼	-0.847*▼	-0.742*▼	-0.706*▼	0.234*▼	-0.025	0.344*▼	0.029	0.450*▼	0.184*▼	0.418*▼
Wass-2	-0.773*▼	-0.847*▼	-0.742*▼	-0.706*▼	0.234*▼	-0.025	0.344*▼	0.029	0.450*▼	0.184*▼	0.418*▼

more advanced approaches to them in later experiments. Second, we need to determine the overall difficulty of predicting transfer performance for each of the three scenarios, such that we know where more advanced solutions are needed. The outcomes of these baseline experiments refine our selection of divergence measures for term distributions, ensuring we include only the most promising ones to use in our feature space for intermediate task selection.

Table 6.1 displays Spearman's ρ correlations for each divergence measure against their relative transfer gain (intermediate-to-target F_1 scores) across the datasets Amazon (AM), Yelp (YL), Yahoo (YH), and TREC-IS (IS). Additionally, results for each dataset are divided by their experimental setting: Zero-shot, Few-shot, and Limited. Significant correlations are marked with an asterisk and the p-value is adjusted for the number of metrics tested using Bonferroni correction ($p < \frac{0.05}{8} = 0.006$). Cosine distance, the baseline measure, is chosen for its widespread use and ease of interpretation. The symbols ▲ and ▼ provide a comparison with the Cosine baseline, enabling us to assess the relative performance of each measure. Values in bold represent the strongest correlations per measure in each distribution group, and those underlined signify the strongest overall correlations per measure. We are primarily concerned with correlation magnitude (from -1 to 1), which describes the strength of association between divergence and relative transfer gain. Following the interpretation of Spearman's correlation magnitudes in Table 2.1, we consider

Spearman's ρ values >=0.7 (a very strong relationship) to be highly effective and likely sufficient for production use. From Table 6.1, we observe the following:

- **There is a clear relationship between domain divergence and transfer learning performance**. Among 176 values reported, 140 (79.6%) are statistically significant. Additionally, 30.1% of correlations are very strong relationships. This substantiates the hypothesis that divergence measures can be used to predict transfer effectiveness. This is a sanity check, demonstrating that the task this thesis addresses is practical.

- **Predictive accuracy in zero-shot settings is high**. Most measures demonstrate very strong correlations in zero-shot scenarios. Both representations are similarly effective, with TF-IDF appearing slightly more so, where every correlation value exceeds our threshold of 0.70. Indeed, using this simple baseline is cheap and should be sufficiently predictive for most common scenarios.

- **Strong performance of information-theoretic measures in zero-shot settings**. Information-theoretic measures exhibit some of the most promising results, particularly in zero-shot settings. Notably, except for the TREC-IS dataset, these measures consistently show the highest correlation values in this setting. Jensen-Shannon divergence demonstrates the strongest correlation amongst all values in the table, at -0.876. This finding strongly indicates that information-theoretic measures are reliable predictors of transfer gain when no target training data is available.

- **Transfer for few-shot and limited settings are difficult to predict**. The absence of statistically significant correlations for Yelp and TREC-IS datasets highlights the challenge in predicting relative transfer gain in few-shot settings. Results in limited settings are also inconsistent. While the correlation magnitudes for Amazon and Yahoo datasets are above 0.4, indicating strong relationships, the variability in the Yelp dataset (with significant values ranging from 0.156 to 0.305) demonstrates the need for more advanced approaches if training in non-zero-shot settings.

The experiments definitively establish a link between domain divergence and transfer learning performance. Information-theoretic and certain geometric divergence measures, particularly Cosine, correlate strongly with relative transfer gain, predominantly in zero-shot contexts. However, their diminished effectiveness in few-shot and limited scenarios requires further exploration of divergence measures for these specific conditions. The findings suggest that while term distributions are valuable in predicting transferability, their effectiveness is significantly influenced by the context, emphasising the need for careful selection of measures aligned with the dataset and training scenario. The following analyses will explore divergence calculation at different levels of abstraction for these distributions

(as detailed in Section 6.2.2), aiming to enhance performance in these challenging settings by comparing subsets of documents within each domain. As baselines, we complement the strong performance of Rényi divergence—selected over Jensen-Shannon divergence and Bhattacharyya distance for its relative consistency—in zero-shot settings with the more consistent performance of Cosine distance in few-shot and limited settings.

6.4.1 Aggregation Analysis

Table 6.2: Spearman's ρ correlations between representation divergence at different levels of representation abstraction and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). *CTD*, *DOM* refer to centroid- and domain-level aggregation, respectively. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of aggregation methods tested ($p < 0.025$). Bold values highlight the strongest correlations within each distribution group. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the highest value among the baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Agg ($\mathcal{D}_I - \mathcal{D}_T$)	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
TF Distributions												
Cos.	$DOM - DOM$	-0.733*	-0.794*	-0.830*	-0.631*	0.350*	0.084	**0.399***	0.024	0.516*	0.293*	**0.491***
Rényi	$DOM - DOM$	**-0.857***	**-0.850***	**-0.863***	**-0.763***	0.180*	-0.048	0.365*	0.012	0.384*	0.156*	0.439*
Cos.	$CTD - DOM$	-0.730*▼	-0.794*▼	-0.773*▼	-0.628*▼	**0.365***▲	0.108	0.396*▼	0.022	**0.532***▲	0.315*▲	0.488*▼
	$CTD - CTD$	-0.740*▼	-0.794*▼	-0.716*▼	-0.624*▼	0.338*▼	0.013	0.307*▼	-0.002	0.503*▼	0.217*▼	0.425*▼
Rényi	$CTD - DOM$	-0.831*▼	-0.835*▼	-0.786*▼	-0.757*▼	0.183*▼	-0.032	0.340*▼	0.026	0.388*▼	0.172*▼	0.435*▼
	$CTD - CTD$	-0.526*▼	-0.744*▼	-0.454*▼	-0.692*▼	-0.002	-0.073	0.064	0.075	0.137*▼	0.051	0.102
CORAL	$CTD - DOM$	-0.264*▼	-0.478*▼	-0.340*▼	-0.014	0.090*▼	-0.039	0.094	-0.087	0.179*▼	0.066	0.157
	$CTD - CTD$	-0.296*▼	-0.455*▼	-0.310*▼	0.045	0.202*▼	**0.184***▲	0.090	-0.058	0.307*▼	0.253*▼	0.144
CMD	$CTD - DOM$	-0.681*▼	-0.779*▼	-0.667*▼	-0.376*▼	0.290*▼	0.103	0.321*▼	0.027	0.448*▼	0.309*▲	0.352*▼
	$CTD - CTD$	-0.681*▼	-0.778*▼	-0.669*▼	-0.377*▼	0.290*▼	0.101	0.323*▼	0.030	0.448*▼	0.307*▲	0.357*▼
MMD-G	$CTD - DOM$	-0.681*▼	-0.779*▼	-0.667*▼	-0.376*▼	0.290*▼	0.103	0.321*▼	0.027	0.448*▼	0.309*▲	0.352*▼
	$CTD - CTD$	-0.681*▼	-0.778*▼	-0.667*▼	-0.377*▼	0.290*▼	0.101	0.322*▼	0.031	0.448*▼	0.307*▲	0.357*▼
MMD-L	$CTD - DOM$	-0.687*▼	-0.778*▼	-0.653*▼	-0.380*▼	0.291*▼	0.107	0.323*▼	0.027	0.448*▼	0.314*▲	0.345*▼
	$CTD - CTD$	-0.697*▼	-0.774*▼	-0.637*▼	-0.378*▼	0.302*▼	0.108	0.338*▼	0.030	0.459*▼	**0.321***▲	0.355*▼
MMD-E	$CTD - DOM$	-0.685*▼	-0.780*▼	-0.656*▼	-0.378*▼	0.291*▼	0.102	0.326*▼	0.028	0.448*▼	0.307*▲	0.354*▼
	$CTD - CTD$	-0.691*▼	-0.775*▼	-0.649*▼	-0.376*▼	0.297*▼	0.105	0.328*▼	0.030	0.454*▼	0.315*▲	0.352*▼
TF-IDF Distributions												
Cos.	$DOM - DOM$	**-0.848***	-0.848*	**-0.829***	**-0.769***	0.260*	-0.012	**0.407***	0.032	0.487*	0.192*	**0.507***
Rényi	$DOM - DOM$	**-0.848***	-0.844*	-0.816*	-0.718*	0.190*	-0.044	0.376*	0.005	0.410*	0.153	0.455*
Cos.	$CTD - DOM$	-0.821*▼	-0.798*▼	-0.717*▼	-0.701*▼	**0.269***▲	0.012	0.386*▼	0.029	**0.498***▲	0.211*▲	0.496*▼
	$CTD - CTD$	-0.812*▼	-0.766*▼	-0.546*▼	-0.661*▼	0.198*▼	**-0.176***▲	0.175	0.051	0.405*▼	-0.025	0.316*▼
Rényi	$CTD - DOM$	-0.741*▼	-0.792*▼	-0.728*▼	-0.678*▼	0.176*▼	-0.040	0.344*▼	0.014	0.383*▼	0.158*▼	0.450*▼
	$CTD - CTD$	-0.285*▼	-0.676*▼	-0.314*▼	-0.601*▼	-0.064*▼	-0.020	0.019	0.022	0.027	0.053	0.047
CORAL	$CTD - DOM$	-0.091*▼	-0.376*▼	-0.294*▼	-0.189*▼	0.130*▼	-0.009	0.110	-0.002	0.190*▼	0.072	0.170
	$CTD - CTD$	-0.133*▼	-0.258*▼	-0.238	-0.188*▼	0.143*▼	-0.138	0.071	-0.035	0.199*▼	**-0.251***▲	0.118
CMD	$CTD - DOM$	-0.772*▼	-0.847*▼	-0.742*▼	-0.710*▼	0.234*▼	-0.022	0.343*▼	0.024	0.450*▼	0.187*▼	0.416*▼
	$CTD - CTD$	-0.770*▼	-0.849*▲	-0.740*▼	-0.708*▼	0.232*▼	-0.026	0.339*▼	0.023	0.448*▼	0.183*▼	0.417*▼
MMD-G	$CTD - DOM$	-0.772*▼	-0.847*▼	-0.742*▼	-0.710*▼	0.234*▼	-0.022	0.344*▼	0.024	0.450*▼	0.187*▼	0.419*▼
	$CTD - CTD$	-0.770*▼	-0.850*▲	-0.739*▼	-0.708*▼	0.233*▼	-0.026	0.342*▼	0.023	0.448*▼	0.183*▼	0.420*▼
MMD-L	$CTD - DOM$	-0.781*▼	-0.840*▼	-0.743*▼	-0.704*▼	0.236*▼	-0.021	0.341*▼	0.023	0.453*▼	0.183*▼	0.413*▼
	$CTD - CTD$	-0.821*▼	-0.850*▲	-0.737*▼	-0.709*▼	0.256*▼	-0.019	0.381*▼	0.022	0.475*▼	0.201*▲	0.439*▼
MMD-E	$CTD - DOM$	-0.778*▼	-0.842*▼	-0.743*▼	-0.705*▼	0.235*▼	-0.021	0.347*▼	0.023	0.452*▼	0.185*▼	0.419*▼
	$CTD - CTD$	-0.807*▼	**-0.851***▲	-0.740*▼	-0.709*▼	0.250*▼	-0.019	0.361*▼	0.023	0.468*▼	0.197*▲	0.428*▼

This analysis extends our foundational results by exploring how divergence calculated at varied abstraction levels can more accurately predict relative transfer gain. Specifically, we compare centroid-domain ($CTD - DOM$) and centroid-centroid ($CTD - CTD$) approaches. We introduce additional divergence measures—CORAL, Central Moment

Discrepancy (CMD), and Maximum Mean Discrepancy (MMD)—which excel in multi-sample comparisons, complementing Cosine distance and Rényi divergence. We contrast the outcomes of $CTD - DOM$ and $CTD - CTD$ with our established domain-domain ($DOM - DOM$) baselines. Our hypothesis posits that centroid-based calculations, aggregating over smaller subsets, may reveal stronger correlations compared to domain-level aggregations. Additionally, we anticipate these new divergence measures will demonstrate competitive performance, particularly in multi-sample scenarios where their unique properties are most advantageous.

In Table 6.2, we extend our analysis to Spearman's ρ correlations at different representation abstraction levels ($CTD - DOM$ and $CTD - CTD$), using Cosine distance and Rényi divergence along with new measures CORAL, CMD, and MMD (with Gaussian, Laplacian, and Energy kernels). This table adopts a similar format to Table 6.1, with significant correlations denoted by an asterisk and a modified p-value adjustment for the number of aggregation methods tested ($p < \frac{0.05}{2} = 0.025$). Aggregation approaches are listed under the Agg ($\mathcal{D}_I - \mathcal{D}_T$) column, with centroid-domain ($CTD - DOM$) and centroid-centroid ($CTD - CTD$) approaches being compared to the domain-domain ($DOM - DOM$) baselines. Bold and underlined values, alongside the ▲ and ▼ symbols, continue to highlight key comparisons relative to the highest value among the baselines (Cosine and Rényi, in grey rows). We are primarily interested in how $CTD - DOM$ and $CTD - CTD$ compare to these baselines. From Table 6.2, we observe:

- **Cosine and Rényi maintain their strongest performance at the domain level**. In the aggregation analysis, Cosine distance and Rényi divergence exhibit their highest efficacy in $DOM - DOM$ comparisons. These measures also demonstrate moderate-to-strong performance in few-shot and limited settings at the $CTD - DOM$ level, though less consistently. Notably, Rényi divergence shows decreased stability in $CTD - CTD$ comparisons, suggesting potential constraints when applying information-theoretic measures to term distributions at the centroid level.

- **CORAL lacks predictive power using term distributions**. The performance of CORAL is notably inconsistent, varying between both negligible and strong correlations in different scenarios. This lack of a predictable pattern undermines its reliability as a measure for predicting transfer learning effectiveness with term distributions.

- **CMD and MMD are competitive, particularly in limited settings**. CMD and MMD demonstrate a range of moderate to very strong correlations in zero-shot settings. Their effectiveness is particularly pronounced in TF-IDF distributions, evidenced by very strong correlations for TREC-IS using TF-IDF, as opposed to

moderate correlations with TF distributions. However, in few-shot and limited settings, their performance is more variable, often not surpassing baseline measures.

In response to RQ1, geometric and information-theoretic measures, particularly Cosine distance and Rényi divergence, emerge as the most effective for predicting relative transfer gain in term distributions. The inconsistent performance of CORAL, CMD, and MMD suggests they don't consistently offer substantial advantages over these baseline measures. In response to RQ2, calculating divergence at various levels of aggregation (such as $CTD - DOM$ and $CTD - CTD$) generally does not yield significant improvements over traditional domain-level aggregation ($DOM - DOM$), refuting our initial hypothesis. While there are instances where $CTD - DOM$ and $CTD - CTD$ offer some benefits, they do not reliably outperform $DOM - DOM$ baselines. Notably, Cosine distance and Rényi divergence are most effective at the domain level, with Rényi demonstrating less stability in $CTD - CTD$ comparisons. The next phase of our experiments will explore the impact of varying vocabulary construction methods on the prediction of relative transfer gain. Building on insights from this and previous experiments, we will continue to focus on the most promising divergence measures and aggregation methods identified thus far: Cosine distance and Rényi divergence, both applied at the domain level.

6.4.2 Vocabulary Configuration Analysis

This experiment focuses on how different vocabulary construction techniques, specifically the Term Ranking Method (TRM) and Domain Context (DC), impact the effectiveness of term distribution representations in predicting relative transfer gain. To control for size-related variables, we maintain a uniform vocabulary size within each datasets. We examine two key aspects: the basis of term selection, either through term frequency (TF) or term frequency-inverse document frequency (TF-IDF) rankings (as part of TRM), and the origin of vocabulary terms, choosing between target domain-specific ($\mathcal{D}_T$) or a combined set from all domains ($\mathcal{D}_I \cup \mathcal{D}_T$) for DC. Our hypothesis posits that TF-IDF, known for its superior ability to identify significant terms compared to TF, will prove more effective for term selection. Furthermore, we hypothesise that a vocabulary centred around the target domain will yield more precise divergence estimates by focusing on term frequency variations important to the target domain, for which we are ranking each of the intermediate tasks.

Table 6.3 extends our analysis to term distribution representations with various vocabulary configurations. This table maintains the format of previous ones, with the same symbols and significance markers (p-value adjusted to $p < \frac{0.05}{4} = 0.013$ for the number of vocabulary configurations). The focus here is on comparing performances across different TRMs and DCs, with key results highlighted in bold and underlined as before. Key

Table 6.3: Spearman's ρ correlations between representation divergence and intermediate-to-target F_1 scores, where each representation is constructed with a different vocabulary configuration. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). TRM and DC denote the Term Ranking Method and the Domain Context, respectively, where $\mathcal{D}_T$ and $\mathcal{D}_I \cup \mathcal{D}_T$ refer to terms included from the target only or from all available domains. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of vocabulary configurations tested ($p < 0.013$). Bold values highlight the strongest correlations within each distribution group. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the highest value among their respective baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

			Zero-shot				Few-shot				Limited		
Measure	TRM	DC	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
TF Distributions													
Cos.	TF	$\mathcal{D}_T$	-0.733*	-0.794*	-0.830*	-0.631*	**0.350***	0.084	**0.399***	0.024	**0.516***	**0.293***	**0.491***
	TF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.791*▲	-0.819*▲	-0.838*▲	-0.629*▼	0.301*▼	0.073	0.394*▼	0.032	0.485*▼	0.280*▼	0.479*▼
	TF-IDF	$\mathcal{D}_T$	-0.733*▬	-0.794*▬	-0.830*▬	-0.631*▬	**0.350***▬	0.084	**0.399***▬	0.024	**0.516***▬	**0.293***▬	**0.491***▬
	TF-IDF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.791*▲	-0.819*▲	-0.838*▲	-0.629*▼	0.301*▼	0.073	0.394*▼	0.032	0.485*▼	0.280*▼	0.479*▼
Rényi	TF	$\mathcal{D}_T$	**-0.857***	-0.850*	**-0.863***	**-0.763***	0.180*	-0.048	0.365*	0.012	0.384*	0.156*	0.439*
	TF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.805*▼	**-0.896***▲	-0.763*▼	-0.521*▼	0.114*▼	0.039	0.355*▼	-0.015	0.358*▼	0.254*▲	0.397*▼
	TF-IDF	$\mathcal{D}_T$	**-0.857***▬	-0.850*▬	**-0.863***▬	**-0.763***▬	0.180*▬	-0.048	0.365*▬	0.012	0.384*▬	0.156*▬	0.439*▬
	TF-IDF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.805*▼	**-0.896***▲	-0.763*▼	-0.521*▼	0.114*▼	0.039	0.355*▼	-0.015	0.358*▼	0.254*▲	0.397*▼
TF-IDF Distributions													
Cos.	TF	$\mathcal{D}_T$	-0.848*	-0.848*	-0.829*	**-0.769***	**0.260***	-0.012	**0.407***	0.032	**0.487***	0.192*	**0.507***
	TF	$\mathcal{D}_I \cup \mathcal{D}_T$	**-0.869***▲	-0.874*▲	**-0.853***▲	-0.737*▼	0.213*▼	-0.017	0.397*▼	0.031	0.452*▼	0.195*▲	0.485*▼
	TF-IDF	$\mathcal{D}_T$	-0.848*▬	-0.848*▬	-0.829*▬	**-0.769***▬	**0.260***▬	-0.012	**0.407***▬	0.032	**0.487***▬	0.192*▬	**0.507***▬
	TF-IDF	$\mathcal{D}_I \cup \mathcal{D}_T$	**-0.869***▲	-0.874*▲	**-0.853***▲	-0.737*	0.213*	-0.017	0.397*	0.031	0.452*	0.195*▲	0.485*▼
Rényi	TF	$\mathcal{D}_T$	-0.848*	-0.844*	-0.816*	-0.718*	0.190*	-0.044	0.376*	0.005	0.410*	0.153	0.455*
	TF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.736*▼	**-0.875***▲	-0.756*▼	-0.471*▼	0.152*▼	0.099	0.401*▲	-0.023	0.396*▼	**0.307***	0.445*▼
	TF-IDF	$\mathcal{D}_T$	-0.848*▬	-0.844*▬	-0.816*▬	-0.718*▬	0.190*▬	-0.044	0.376*▬	0.005	0.410*▬	0.153	0.455*▬
	TF-IDF	$\mathcal{D}_I \cup \mathcal{D}_T$	-0.736*▼	**-0.875***▲	-0.756*▼	-0.471*▼	0.152*▼	0.099	0.401*▲	-0.023	0.396*▼	**0.307***	0.445▼*

findings include:

- **There is no significant difference between TRMs**. The results reveals that the choice of TRM, whether TF or TF-IDF, does not significantly impact divergence outcomes. Across all datasets and settings, the variation in TRM, regardless of the domain context, leads to identical correlation values. This observation suggests that both methods are equally effective in identifying important terms. Therefore, we recommend selecting a TRM that aligns with the distribution type, primarily to optimise computational efficiency.

- **DC matters, but is context-dependent**. The influence of DC is prominent but varies across settings. In zero-shot scenarios, the Cosine distance typically exhibits stronger correlations with vocabularies encompassing a broader range of terms from all domains ($\mathcal{D}_I \cup \mathcal{D}_T$), with TREC-IS being an exception. Conversely, in few-shot and limited settings, a target-specific vocabulary ($\mathcal{D}_T$) tends to produce better correlations. Rényi divergence consistently shows a preference for target-focused vocabularies across most datasets and settings. Given that target-focused vocabularies result in the best overall Cosine distance correlations in few-shot and limited settings, where even moderate increases in performance are important, we recommend

adopting DC=$\mathcal{D}_T$ for this measure.

In response to RQ3, the results indicate that TRM has a minimal impact on the representational quality of term distributions, suggesting that both TF and TF-IDF are viable options for vocabulary construction. In contrast, the choice of DC significantly influences the predictive power of each distribution.

It is worth noting that our experiments use local IDFs derived from in-domain data when constructing vocabularies based on TF-IDF rankings. While using global IDFs from a larger, cleaner collection like Wikipedia has the potential to better capture term importance, our DC results provide strong evidence that target-focused vocabularies (and thereby, the combination of target-focused IDFs and target-focused vocabularies) are more predictive of transfer learning success.

In terms of divergence measures, while Cosine distance favours more variety in terms from all domains in zero-shot settings, the overarching trend indicates the effectiveness of target-focused vocabularies, especially in few-shot and limited contexts. Rényi divergence consistently aligns with target-specific vocabularies, showing notable improvement in zero-shot scenarios with TF distributions.

In summary, these findings suggest that for future analyses, vocabularies should be constructed from the target domain, with TRM selection based on computational considerations—TF for TF distributions and TF-IDF for TF-IDF distributions.

6.5 Linguistic Feature Distributions Evaluation

Similar to the baseline experiments in Section 6.4, the aim of these experiments are to: (1) evaluate the base effectiveness of Linguistic Feature Distributions (LFD)—Named Entity (NE), Linguistic Dependency (DEP), and Universal/Extended Part-of-Speech (U/XPOS) distributions—with relative transfer gain and (2) identify where more advanced solutions, based on aggregation analyses, are needed. We hypothesise that linguistic feature distributions, capturing more complex syntactic and semantic structures within text, may be able to adapt to areas where Term Distributions (TD) fell short, i.e. in few-shot and limited experimental settings. Furthermore, LFDs are categorical and thus occupy a significantly lower-dimensional space than the high-dimensional space characteristic of TDs. This reduced dimensionality may enhance divergence estimations, given that certain measures exhibit limitations when applied to high-dimensional, sparse data.

Table 6.4 presents an analysis of the Spearman's ρ correlations between LFD divergence and relative transfer gain (intermediate-to-target F_1 performance). The table maintains the same formatting as in Section 6.4, with different distribution types divided into separate sections of the table. We use the same measures, symbols, and significance markers

Table 6.4: Spearman's ρ correlations between domain (DOM) representation divergence and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of measures tested ($p < 0.006$). Bold values highlight the strongest correlations within each distribution group. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the best results to their respective cosine (Cos.) baselines (grey row) within each measure, representing higher or lower correlation magnitudes, respectively.

	Zero-shot				Few-shot				Limited		
Measure	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
NE Distributions											
Cos.	-0.789*	-0.653*	-0.466*	-0.576*	0.145*	-0.148	0.218	0.111	0.350*	0.036	0.300*
L_1	-0.813*▲	-0.695*▲	-0.471*▲	-0.586*▲	0.159*▲	-0.116	0.231	0.089	0.369*▲	0.059	0.303*▲
L_2	-0.791*▲	-0.658*▲	-0.475*▲	-0.564*▼	**0.165***▲	-0.154	0.241	0.087	0.367*▲	0.056	0.315*▲
JS	**-0.818***▲	-0.714*▲	-0.475*▲	**-0.641***▲	0.159*▲	-0.095	0.239	0.108	**0.375***▲	0.088	0.320*▲
Rényi	-0.811*▲	-0.688*▲	-0.472*▲	-0.615*▲	0.121*▼	-0.070	0.256	0.130	0.338*▼	0.123	**0.329***▲
Bhat.	**-0.818***▲	**-0.715***▲	**-0.478***▲	-0.639*▲	0.158*▲	-0.093	0.244	0.110	**0.375***▲	0.089	0.327*▲
Wass-1	-0.791*▲	-0.658*▲	-0.475*▲	-0.564*▼	**0.165***▲	-0.154	0.241	0.087	0.367*▲	0.056	0.315*▲
Wass-2	-0.791*▲	-0.658*▲	-0.475*▲	-0.564*▼	**0.165***▲	-0.154	0.241	0.087	0.367*▲	0.056	0.315*▲
DEP Distributions											
Cos.	-0.723*▼	-0.706*▼	-0.415*▼	-0.558*▼	0.263*▼	-0.073	0.159	-0.011	0.431*▼	0.073	0.119
L_1	-0.746*▲	-0.713*▲	-0.366*▼	-0.606*▲	0.261*▼	-0.083	0.142	-0.021	0.434*▲	0.060	0.110
L_2	-0.725*▲	-0.706*–	-0.396*▼	-0.556*▼	**0.266***▲	-0.075	0.149	-0.026	0.433*▲	0.068	0.110
JS	-0.778*▲	-0.719*▲	-0.448*▲	-0.643*▲	0.251*▼	-0.104	0.211	-0.004	**0.437***▲	0.063	0.163
Rényi	**-0.779***▲	**-0.720***▲	-0.446*▲	**-0.647***▲	0.244*▼	-0.109	0.198	-0.006	0.430*▼	0.056	0.151
Bhat.	-0.778*▲	-0.719*▲	**-0.449***▲	-0.643*▲	0.251*▼	-0.104	0.212	-0.004	**0.437***▲	0.063	0.164
Wass-1	-0.725*▲	-0.706*▼	-0.396*▼	-0.556*▼	**0.266***▲	-0.075	0.149	-0.026	0.433*▲	0.068	0.110
Wass-2	-0.725*▲	-0.706*▼	-0.396*▼	-0.556*▼	**0.266***▲	-0.075	0.149	-0.026	0.433*▲	0.068	0.110
UPOS Distributions											
Cos.	-0.725*▼	-0.699*▼	-0.392*▼	-0.542*▼	0.221*▼	-0.112	0.181	-0.014	0.397*▼	0.048	0.125
L_1	-0.728*▲	**-0.705***▲	-0.370*▼	-0.584*▲	**0.233***▲	-0.099	0.174	-0.007	**0.406***▲	0.050	0.124
L_2	-0.727*▲	-0.699*–	-0.390*▼	-0.553*▲	0.219*▼	-0.113	0.180	-0.019	0.395*▼	0.047	0.122
JS	-0.804*▲	-0.703*▲	**-0.487***▲	**-0.617***▲	0.196*▼	-0.107	0.246	-0.002	0.400*▲	0.052	0.178
Rényi	**-0.806***▲	-0.703*▲	-0.483*▲	-0.616*▲	0.172*▼	-0.106	0.233	-0.004	0.377*▼	0.054	0.162
Bhat.	-0.804*▲	-0.703*▲	-0.484*▲	**-0.617***▲	0.196*▼	-0.107	0.247	-0.002	0.400*▲	0.052	0.178
Wass-1	-0.727*▲	-0.699*–	-0.390*▼	-0.553*▲	0.219*▼	-0.113	0.180	-0.019	0.395*▼	0.047	0.122
Wass-2	-0.727*▲	-0.699*–	-0.390*▼	-0.553*▲	0.219*▼	-0.113	0.180	-0.019	0.395*▼	0.047	0.122
XPOS Distributions											
Cos.	-0.757*▼	-0.686*▼	-0.455*▼	-0.590*▼	0.230*▼	-0.091	0.215	0.001	0.415*▼	0.081	0.174
L_1	-0.764*▲	**-0.694***▲	-0.439*▼	-0.645*▲	**0.245***▲	-0.097	0.212	-0.005	0.431*▲	0.065	0.171
L_2	-0.759*▲	-0.684*▼	-0.441*▼	-0.616*▲	0.226*▼	-0.086	0.217	-0.010	0.412*▼	0.075	0.168
JS	**-0.816***▲	-0.687*▲	-0.541*▲	-0.667*▲	0.234*▲	-0.115	0.267	-0.012	0.437*▲	0.069	0.221
Rényi	**-0.816***▲	-0.687*▲	**-0.542***▲	-0.661*▲	0.212*▼	-0.119	0.252	-0.010	0.415*–	0.066	0.199
Bhat.	**-0.816***▲	-0.687*▲	**-0.542***▲	**-0.668***▲	0.234*▲	-0.116	0.269	-0.012	**0.438***▲	0.068	0.222
Wass-1	-0.759*▲	-0.684*▼	-0.441*▼	-0.616*▲	0.226*▼	-0.086	0.217	-0.010	0.412*▼	0.075	0.168
Wass-2	-0.759*▲	-0.684*▼	-0.441*▼	-0.616*▲	0.226*▼	-0.086	0.217	-0.010	0.412*▼	0.075	0.168

(p-value adjusted to $p < \frac{0.05}{8} = 0.006$ to correct for the number of measures tested for each distribution). From Table 6.4, we make the following observations:

- **Strong correlations in zero-shot settings across all distributions**. Similar to the findings for TD baselines, we observe strong (0.40-0.69) to very strong (≥ 0.70) correlation magnitudes in zero-shot settings for the majority of measures. However, l_p-norm and Wasserstein distances exhibit inconsistencies, particularly for DEP and UPOS distributions within the Yahoo dataset, where they demonstrate a moderate correlation with performance (-0.366 to -0.390).

- **Information-theoretic measures are highly effective in zero-shot settings**. Information-theoretic measures consistently achieved the strongest correlations for zero-shot settings. Bhattacharyya distance emerged as the best-performing information-theoretic measure in this scenario, achieving the strongest of all zero-shot measures for 3 out of 4 datasets. For specific distributions, the results are as follows: (1) Bhattacharyya distance (Bhat.) is the best divergence measure for NE distributions, achieving 3 out of 4 of the best within-distribution scores (bold) and the correlations for Bhat. range between -0.639 for TREC-IS to -0.818 for Amazon; (2) Rényi divergence is the best measure for use with DEP distributions, achieving 2 out of 4 of the best within-distribution scores, with Rényi's correlations existing in the range -0.446 to -0.779; (3) Jensen-Shannon divergence (JS) is the best measure for the Universal Tag variant of our POS distributions, with correlations between -0.487 and -0.804 and achieving 2 out of 4 of the best within-distribution scores; and (4) Bhattacharyya distance is the best measure for the Extended Tag variant of our POS distributions, ranging from -0.542 (Yahoo) to -0.816 (Amazon) and achieving 3 out of 4 of the best within-distribution scores.

- **Few-shot settings are difficult to predict**. Yelp, Yahoo, and TREC-IS datasets report no statistically significant correlations in few-shot settings while Amazon's significant correlations range from negligible (< 0.20) to weak ($0.20 - 0.29$). This represents a significant decrease from the performance range and number of significant values for TDs, where we have moderate-to-strong correlations for Yahoo and weak-to-moderate for Amazon. These results demonstrate that LFDs struggle to capture meaningful relationships in difficult experimental settings.

- **Different datasets "prefer" different distribution types**. NE and DEP distributions are particularly effective for Amazon in zero- and few-shot settings, the former achieving the best overall scores in zero-shot and the latter achieving the best overall scores in few-shot settings. Yelp generally achieves a higher number of very strong correlations using DEP distributions in zero-shot. In few-shot and limited settings, Yelp reports no statistically significant correlations. Yahoo generally achieves its best zero-shot results using the XPOS and NE distributions. NE distributions are also the only dataset to achieve statistically significant (moderate) correlations with the Yahoo dataset, suggesting a high presence of named entities in their domains. Similarly to Yelp, Yahoo reports no statistically significant correlations in few-shot settings.

In response to RQ4, the experiments indicate that information-theoretic measures, particularly Bhattacharyya distance, are most effective in capturing the correlation between LFDs and relative transfer gain. Their consistent performance in zero-shot settings high-

lights their utility in predicting the effectiveness when no target training data is available. The results also show that LFDs struggle, generally, in few-shot and limited settings. Out of 224 correlation values, only 32.1% are statistically significant. These scores provide the justification for exploring the effect of calculating divergence at different levels of abstraction for these representations. We will use the best measure from these experiments, Bhattacharyya distance, as the baseline for our experiments.

6.5.1 Aggregation Analysis

Table 6.5: Spearman's ρ correlations between NE representation divergence at different levels of representation abstraction and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). *CTD*, *DOM* refer to centroid- and domain-level aggregation, respectively. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of aggregation levels tested ($p < 0.013$). Bold values highlight the strongest correlations within each measure. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the highest value among the baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Agg ($\mathcal{D}_I - \mathcal{D}_T$)	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Bhat.	$DOM - DOM$	**-0.818***	**-0.715*▲**	**-0.478***	**-0.639***	0.158*	-0.093	0.244	0.110	**0.375***	0.089	**0.327***
	$INST - DOM$	0.165*▼	0.066	0.042	-0.074	0.121*▼	0.070	0.105	0.051	0.073*▼	0.046	0.089
	$CTD - DOM$	-0.812*▼	-0.709*▼	-0.462*▼	-0.614*▼	0.156*▼	-0.142	0.218	0.099	0.369*▼	0.016	0.301*▼
	$INST - INST$	0.269*▼	0.014	0.119	-0.182*▼	**-0.175*▲**	-0.021	0.019	0.113	-0.203*▼	**-0.276*▲**	-0.095
	$CTD - CTD$	-0.806*▼	-0.687*▼	-0.447*▼	-0.638*▼	0.150*▼	-0.101	0.229	0.117	0.365*▼	0.056	0.293*▼
CORAL	$INST - DOM$	0.189*▼	-0.224*▼	-0.241	0.030	0.033	-0.147	-0.064	0.062	0.012	-0.067	-0.042
	$CTD - DOM$	0.260*▼	-0.122	0.004	0.101	0.050	-0.084	0.060	0.010	-0.004	-0.047	0.009
	$INST - INST$	0.175*▼	**-0.257*▼**	**-0.344*▼**	-0.050	-0.009	0.117	**0.351*▲**	0.053	-0.016	**0.357*▲**	**0.381*▲**
	$CTD - CTD$	**0.302*▼**	-0.165*▼	-0.008	0.088	**-0.160*▼**	0.035	0.183	0.029	**-0.199*▼**	0.127	0.019
CMD	$INST - DOM$	-0.590*▼	-0.588*▼	-0.380*▼	-0.333*▼	**0.232*▲**	-0.137	**0.320*▲**	0.090	0.387*▲	0.125	**0.394*▲**
	$CTD - DOM$	**-0.791*▼**	-0.655*▼	-0.478*▬	-0.556*▼	0.164*▲	-0.156*▲	0.244	0.086	0.367*▼	0.055	0.318*▼
	$INST - INST$	-0.790*▼	**-0.673*▼**	**-0.512*▼**	**-0.682*▲**	0.192*▲	-0.155*▲	0.275*▲	0.022	**0.398*▲**	0.051	0.349*▲
	$CTD - CTD$	-0.790*▼	-0.654*▼	-0.482*▲	-0.572*▼	0.168*▲	**-0.162*▲**	0.257*▲	0.080	0.370*▼	0.048	0.327*▬
MMD-G	$INST - DOM$	-0.678*▼	-0.532*▼	-0.427*▼	-0.306*▼	0.210*▲	-0.098	**0.391*▲**	0.085	0.372*▼	0.149	**0.469*▲**
	$CTD - DOM$	**-0.791*▼**	-0.655*▼	-0.477*▼	-0.557*▼	0.164*▲	-0.156*▲	0.245	0.086	0.366*▼	0.055	0.318*▼
	$INST - INST$	-0.779*▼	**-0.675*▼**	**-0.574*▲**	**-0.676*▲**	**0.215*▲**	-0.149	0.287*▲	0.002	**0.408*▲**	0.090	0.398*▲
	$CTD - CTD$	-0.790*▼	-0.654*▼	-0.484*▲	-0.572*▼	0.169*▲	**-0.161*▲**	0.255*▲	0.080	0.370*▼	0.050	0.330*▲
MMD-L	$INST - DOM$	0.200*▼	-0.082	-0.013	-0.103	0.106*▼	0.048	0.126	0.045	0.050	0.113	0.092
	$CTD - DOM$	**-0.777*▼**	**-0.662*▼**	-0.520*▲	-0.559*▼	0.094*▼	**-0.159*▲**	0.205	0.079	0.296*▼	0.046	0.264*▼
	$INST - INST$	-0.709*▼	-0.638*▼	-0.543*▲	**-0.667*▲**	**0.192*▲**	-0.113	**0.309*▲**	0.038	0.377*▲	0.087	0.301*▼
	$CTD - CTD$	-0.772*▼	-0.659*▼	**-0.554*▲**	-0.559*▼	0.187*▲	-0.150	0.212	0.065	**0.378*▲**	0.062	**0.359*▲**
MMD-E	$INST - DOM$	-0.680*▼	-0.561*▼	-0.445*▼	-0.388*▼	0.205*▲	-0.085	**0.374*▲**	0.091	0.365*▼	**0.177*▲**	0.450*▲
	$CTD - DOM$	**-0.793*▼**	-0.658*▼	-0.489*▲	-0.559*▼	0.160*▲	**-0.158*▲**	0.248	0.082	0.362*▼	0.053	0.324*▼
	$INST - INST$	-0.782*▼	**-0.664*▼**	**-0.568*▲**	**-0.671*▲**	**0.210*▲**	-0.157*▲	0.289*▲	0.004	**0.406*▲**	0.078	**0.394*▲**
	$CTD - CTD$	-0.790*▼	-0.659*▼	-0.496*▲	-0.570*▼	0.176*▲	-0.157*▲	0.259*▲	0.073	0.375*▬	0.054	0.341*▲

In this aggregation analysis for Linguistic Feature Distributions (LFDs), we investigate how divergence calculated at varied abstraction levels can more accurately predict relative transfer gain for LFDs. Specifically, we compare instance-domain ($INST - DOM$), centroid-domain ($CTD - DOM$), instance-to-instance ($INST - INST$), and centroid-centroid ($CTD - CTD$) approaches. Similarly to Section 6.4.1, we introduce CORAL, Central Moment Discrepancy (CMD) and Maximum Mean Discrepancy (MMD) measures as they excel in multi-sample comparisons due to their inherent properties. The choice to include instance-based calculations is due to the reduced dimensionality of LFDs—

compared with Term Distributions (TD)—which makes them much less computationally expensive to compute. This reduced dimensionality allows for better comparisons to be made for both instance- and centroid-based divergence and hence, we anticipate improved performance over domain-aggregated comparisons for these distributions. Due to the number of comparisons being made, the aggregation analysis for each LFD is divided into separate tables, each with a similar format. Each table presents Spearman's ρ correlations between representation divergence at different levels of abstraction and transfer learning performance, where the threshold for significance is corrected for the number of aggregation methods tested ($p < \frac{0.05}{4} = 0.013$).

Table 6.5 presents the results of the aggregation analysis for Named Entity (NE) distributions. The table follows a similar format to the table for the aggregation analysis of TDs (Section 6.4.1). Aggregation approaches are listed under the "Agg ($\mathcal{D}_I - \mathcal{D}_T$)" column, with instance-instance ($INST - INST$), instance-domain ($INST - DOM$), centroid-domain ($CTD - DOM$), and centroid-centroid ($CTD - CTD$) approaches being compared to the domain-domain ($DOM - DOM$) Bhattacharyya distance baseline. From Table 6.5, we observe:

- **Bhattacharyya is not effective at the instance level**. Bhattacharyya distance scores fall sharply from the $DOM - DOM$ aggregation to either $INST - DOM$ or $INST - INST$ aggregation levels, going from strong/very strong correlations to either negligible or weak correlations with relative transfer gain. The performance of centroid-based approaches is close to but fails to beat the domain-level baseline.

- **CORAL lacks sufficient predictive power**. CORAL displays inconsistent performance, suggesting its limited utility for NE distributions, with variable results across datasets and settings.

- **Instance- and centroid- based comparisons are effective for CMD, MMD measures**. Within the CMD and MMD metrics, $CTD - DOM$ aggregations prove most effective in the Amazon dataset's zero-shot setting. A consistent pattern is observed across CMD and MMD measures: $INST - INST$ aggregations consistently show the strongest correlations compared to other aggregation methods. For the 9 dataset/setting combinations where these metrics have significant correlations, $INST - INST$ emerges as the superior aggregation approach within each measure in the following instances: (1) CMD leads in 4 out of 9 cases, (2) MMD with Gaussian kernel excels in 5 out of 9 cases, (3) MMD with Laplacian kernel is top-performing in 3 out of 9 cases (tied with both $CTD - DOM$ and $CTD - CTD$), (4) MMD with Energy kernel is most effective in 6 out of 9 cases. This pattern suggests that instance- and centroid-based aggregation calculations can enhance performance,

especially for distributions that are not characterised by high dimensionality and sparsity, as is often the case with term distributions.

- **Few-shot and limited settings are difficult to predict, but show signs of improvement**. Predicting outcomes in few-shot and limited settings remains a complex task, as no single measure consistently enhances performance across all these scenarios. However, there are indications of progress, particularly in the Yahoo dataset for few-shot settings. Here, CMD and MMD metrics demonstrate notable improvements. This contrasts with earlier baseline experiments where Yahoo in the few-shot setting yielded no significant results. Now, we observe moderate correlations with performance using these measures. Additionally, in few-shot and limited settings for Yelp, our aggregation analysis reveals slight yet statistically significant improvements, with correlations ranging from negligible to moderate, albeit with no consistent pattern.

For NE distributions, Bhattacharyya distance shows a marked decline in performance when moving from domain-level to instance-level aggregations, and CORAL demonstrates inconsistent results, casting doubts on their utility for NE distributions. Meanwhile, CMD and MMD measures provide promising results, particularly in instance- and centroid-based comparisons, with $INST - INST$ aggregations often emerging as the most effective. Furthermore, while no single measure uniformly enhances performance in few-shot and limited settings, CMD and MMD show signs of improvement in difficult settings, especially in the Yahoo dataset for few-shot scenarios and Yelp in limited settings. Ultimately, we find that divergence results using Bhattacharyya distance with the $DOM-DOM$ aggregation, CMD at the $INST - INST$ aggregation level, and MMD (Gaussian kernel), also at the $INST - INST$ aggregation level, are suitable candidates for use as features in our task selection experiments in later chapters.

Table 6.6 presents the aggregation analysis results for Linguistic Dependency (DEP) distributions, using a format consistent with our previous analysis. The observations from Table 6.6 reveal parallels to the findings in the NE distribution analysis, specifically: (1) Bhattacharyya distance's decreased performance in instance-based comparisons; (2) CORAL's varying effectiveness across different datasets and settings; and (3) the effectiveness of of instance- and centroid-based comparisons for CMD and MMD measures. Table 6.6 also shows some unique aspects specific to DEP distributions, namely:

- **Yahoo is difficult to predict in few-shot and limited settings**. Contrasting with the NE distribution findings, Yahoo shows no significant correlations in few-shot and limited settings for DEP distributions, underscoring the limitations of DEP distributions in capturing domain characteristics in these scenarios.

- **Consistent performance of instance-instance approach for all MMD measures**. Across all statistically significant cases, with the exception of Amazon in zero-shot, MMD measures, particularly with Gaussian kernel, consistently favour the $INST - INST$ aggregation method, often surpassing baseline performances.

Overall, while DEP distributions echo many trends seen in NE distributions, they exhibit a notable performance dip in Yahoo's few-shot and limited settings. This suggests that DEP distributions may be less effective at revealing domain characteristics in the specific question/answer language style of this dataset. Meanwhile, $INST - INST$ aggregation continues to be a robust approach for CMD and MMD measures, with MMD showing particular alignment with this method. In selecting measures for task selection using DEP distributions, we follow the same recommendations as with NE distributions and use Bhattacharyya distance ($DOM - DOM$), CMD ($INST - INST$), and MMD-G ($INST - INST$).

Table 6.7 presents the aggregation analysis results for the universal and extended tag variants of our part-of-speech distributions (UPOS and XPOS), using a format similar to our previous analysis, but with both distributions condensed into a single table due

Table 6.6: Spearman's ρ correlations between DEP representation divergence at different levels of representation abstraction and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). CTD, DOM refer to centroid- and domain-level aggregation, respectively. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of aggregation levels tested ($p < 0.013$). Bold values highlight the strongest correlations within each measure. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the highest value among the baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Agg ($\mathcal{D}_I - \mathcal{D}_T$)	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Bhat.	$DOM - DOM$	**-0.778***	**-0.719***	-0.449*	-0.643*	**0.251***	-0.104	0.212	-0.004	**0.437***	0.063	0.164
	$INST - DOM$	0.171*$_{\blacktriangledown}$	0.148	-0.098	-0.138	0.063*$_{\blacktriangledown}$	**0.231***$_{\blacktriangle}$	0.089	0.037	0.049	0.225*$_{\blacktriangle}$	0.116
	$CTD - DOM$	-0.763*$_{\blacktriangledown}$	-0.703*$_{\blacktriangledown}$	**-0.452***$_{\blacktriangle}$	-0.637*$_{\blacktriangledown}$	0.250*$_{\blacktriangledown}$	-0.103	0.208	-0.008	0.433*$_{\blacktriangledown}$	0.057	0.166
	$INST - INST$	0.285*$_{\blacktriangledown}$	-0.168*$_{\blacktriangledown}$	-0.126	-0.266*$_{\blacktriangledown}$	-0.075*$_{\blacktriangledown}$	-0.035	-0.093	0.023	-0.095*$_{\blacktriangledown}$	**-0.231***$_{\blacktriangle}$	-0.088
	$CTD - CTD$	-0.748*$_{\blacktriangledown}$	-0.692*$_{\blacktriangledown}$	**-0.453***$_{\blacktriangle}$	**-0.645***$_{\blacktriangle}$	0.243*$_{\blacktriangledown}$	-0.114	0.212	-0.036	0.419*$_{\blacktriangledown}$	0.030	0.171
CORAL	$INST - DOM$	0.188*$_{\blacktriangledown}$	**0.311***$_{\blacktriangledown}$	0.017	0.145	0.019	**0.211***$_{\blacktriangle}$	0.029	0.008	0.002	0.092	0.037
	$CTD - DOM$	0.124*$_{\blacktriangledown}$	0.299*$_{\blacktriangledown}$	0.056	0.081	0.003	0.203*$_{\blacktriangle}$	0.017	-0.026	-0.004	0.085	0.043
	$INST - INST$	**0.221***$_{\blacktriangledown}$	0.163*$_{\blacktriangledown}$	-0.047	0.075	-0.018	-0.032	-0.098	0.065	-0.027	**-0.351***$_{\blacktriangle}$	-0.063
	$CTD - CTD$	0.097*$_{\blacktriangledown}$	0.194*$_{\blacktriangledown}$	-0.137	**0.174***$_{\blacktriangledown}$	0.007	-0.014	0.068	-0.037	-0.001	-0.161*$_{\blacktriangle}$	0.099
CMD	$INST - DOM$	-0.353*$_{\blacktriangledown}$	-0.566*$_{\blacktriangledown}$	-0.350*$_{\blacktriangledown}$	-0.334*$_{\blacktriangledown}$	0.201*$_{\blacktriangledown}$	0.030	0.095	0.004	0.283*$_{\blacktriangledown}$	**0.189***$_{\blacktriangle}$	0.069
	$CTD - DOM$	**-0.725***$_{\blacktriangledown}$	-0.706*$_{\blacktriangledown}$	-0.398*$_{\blacktriangledown}$	-0.552*$_{\blacktriangledown}$	0.266*$_{\blacktriangle}$	-0.073	0.150	-0.022	0.433*$_{\blacktriangledown}$	0.071	0.110
	$INST - INST$	-0.719*$_{\blacktriangledown}$	**-0.713***$_{\blacktriangledown}$	**-0.467***$_{\blacktriangle}$	-0.531*$_{\blacktriangledown}$	**0.296***$_{\blacktriangle}$	-0.053	0.161	-0.019	**0.464***$_{\blacktriangle}$	0.083	0.135
	$CTD - CTD$	-0.722*$_{\blacktriangledown}$	-0.706*$_{\blacktriangledown}$	-0.407*$_{\blacktriangledown}$	**-0.571***$_{\blacktriangledown}$	0.263*$_{\blacktriangle}$	-0.074	0.162	-0.033	0.429*$_{\blacktriangledown}$	0.069	0.127
MMD-G	$INST - DOM$	-0.284*$_{\blacktriangledown}$	-0.560*$_{\blacktriangledown}$	-0.350*$_{\blacktriangledown}$	-0.373*$_{\blacktriangledown}$	0.211*$_{\blacktriangledown}$	0.028	0.101	-0.006	0.279*$_{\blacktriangledown}$	**0.178***$_{\blacktriangle}$	0.079
	$CTD - DOM$	**-0.725***$_{\blacktriangledown}$	-0.706*$_{\blacktriangledown}$	-0.398*$_{\blacktriangledown}$	-0.553*$_{\blacktriangledown}$	0.266*$_{\blacktriangle}$	-0.073	0.150	-0.022	0.433*$_{\blacktriangledown}$	0.070	0.110
	$INST - INST$	**-0.725***$_{\blacktriangledown}$	**-0.717***$_{\blacktriangledown}$	**-0.470***$_{\blacktriangle}$	**-0.596***$_{\blacktriangledown}$	**0.297***$_{\blacktriangle}$	-0.057	0.172	-0.014	**0.462***$_{\blacktriangle}$	0.082	0.151
	$CTD - CTD$	-0.722*$_{\blacktriangledown}$	-0.706*$_{\blacktriangledown}$	-0.405*$_{\blacktriangledown}$	-0.573*$_{\blacktriangledown}$	0.263*$_{\blacktriangle}$	-0.074	0.162	-0.031	0.429*$_{\blacktriangledown}$	0.069	0.126
MMD-L	$INST - DOM$	0.011	-0.454*$_{\blacktriangledown}$	-0.259*$_{\blacktriangledown}$	-0.188*$_{\blacktriangledown}$	0.161*$_{\blacktriangledown}$	0.025	0.111	-0.044	0.181*$_{\blacktriangledown}$	0.047	0.123
	$CTD - DOM$	-0.729*$_{\blacktriangledown}$	-0.709*$_{\blacktriangledown}$	-0.394*$_{\blacktriangledown}$	-0.574*$_{\blacktriangledown}$	0.267*$_{\blacktriangle}$	-0.076	0.142	-0.018	0.435*$_{\blacktriangledown}$	0.067	0.101
	$INST - INST$	-0.690*$_{\blacktriangledown}$	**-0.727***$_{\blacktriangle}$	**-0.453***$_{\blacktriangle}$	**-0.643***$_{\blacktriangle}$	**0.283***$_{\blacktriangle}$	-0.060	0.217	-0.032	**0.443***$_{\blacktriangle}$	0.102	0.195
	$CTD - CTD$	**-0.733***$_{\blacktriangledown}$	-0.710*$_{\blacktriangledown}$	-0.396*$_{\blacktriangledown}$	-0.590*$_{\blacktriangledown}$	0.263*$_{\blacktriangle}$	-0.071	0.149	-0.018	0.431*$_{\blacktriangledown}$	0.076	0.111
MMD-E	$INST - DOM$	-0.069*$_{\blacktriangledown}$	-0.482*$_{\blacktriangledown}$	-0.293*$_{\blacktriangledown}$	-0.302*$_{\blacktriangledown}$	0.192*$_{\blacktriangledown}$	0.021	0.094	-0.018	0.225*$_{\blacktriangledown}$	0.090	0.084
	$CTD - DOM$	**-0.728***$_{\blacktriangledown}$	-0.708*$_{\blacktriangledown}$	-0.400*$_{\blacktriangledown}$	-0.568*$_{\blacktriangledown}$	0.267*$_{\blacktriangle}$	-0.076	0.145	-0.018	0.435*$_{\blacktriangledown}$	0.067	0.106
	$INST - INST$	-0.718*$_{\blacktriangledown}$	**-0.721***$_{\blacktriangle}$	**-0.466***$_{\blacktriangle}$	**-0.607***$_{\blacktriangledown}$	**0.295***$_{\blacktriangle}$	-0.061	0.179	-0.016	**0.459***$_{\blacktriangle}$	0.088	0.159
	$CTD - CTD$	-0.730*$_{\blacktriangledown}$	-0.710*$_{\blacktriangledown}$	-0.402*$_{\blacktriangledown}$	-0.586*$_{\blacktriangledown}$	0.263*$_{\blacktriangle}$	-0.071	0.150	-0.022	0.431*$_{\blacktriangledown}$	0.076	0.115

Table 6.7: Spearman's ρ correlations between UPOS and XPOS representation divergence at different levels of representation abstraction and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). *CTD*, *DOM* refer to centroid- and domain-level aggregation, respectively. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of aggregation levels tested ($p < 0.013$). Bold values highlight the strongest correlations within each measure. Best overall score for each dataset/setting is underlined. Symbols ▲ and ▼ compare values to the highest value among the baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Agg ($\mathcal{D}_I - \mathcal{D}_T$)	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
UPOS Distributions												
Bhat.	*DOM − DOM*	**-0.804***	**-0.703***	-0.484*	-0.617*	**0.196***	-0.107	0.247	-0.002	**0.400***	0.052	0.178
	INST − DOM	0.158*▼	0.045	-0.233	-0.192*▼	0.051	0.143	0.153	0.032	0.045	0.079	0.175
	CTD − DOM	-0.802*▼	-0.693*▼	**-0.495***▲	-0.596*▼	**0.196***▬	-0.111	0.249	0.001	**0.400***▬	0.042	0.185
	INST − INST	0.280*▼	-0.123	-0.202	-0.214*▼	-0.089*▼	-0.071	-0.019	0.018	-0.103*▼	**-0.293***▲	-0.010
	CTD − CTD	-0.796*▼	-0.692*▼	-0.492*▲	**-0.619***▲	0.191*▼	-0.109	**0.252***▲	-0.031	0.392*▼	0.045	0.190
CORAL	*INST − DOM*	0.197*▼	0.256*▼	0.008	**0.201***▼	0.014	0.158*▲	0.006	0.007	0.003	0.066	0.050
	CTD − DOM	0.117*▼	**0.269***▼	0.059	0.109	0.003	0.180*▲	0.019	-0.045	0.005	0.069	0.058
	INST − INST	**0.226***▼	0.177*▼	-0.080	0.168*▼	-0.029	-0.106	-0.056	0.064	-0.046	-0.410*▼	-0.006
	CTD − CTD	0.065*▼	0.241*▼	-0.134	0.172*▼	**0.068***▼	**-0.213***▲	0.036	-0.132	**0.077***▼	**-0.374***▲	0.094
CMD	*INST − DOM*	-0.409*▼	-0.568*▼	-0.412*▼	-0.345*▼	0.027	0.023	0.131	0.035	0.128*▼	**0.199***▼	0.082
	CTD − DOM	**-0.727***▼	-0.699*▼	-0.392*▼	-0.546*▼	0.218*▲	-0.113	0.179	-0.019	0.394*▼	0.047	0.122
	INST − INST	-0.692*▼	**-0.709***▲	**-0.506***▲	-0.529*▼	**0.259***▲	-0.092	0.216	-0.024	**0.423***▲	0.061	0.185
	CTD − CTD	-0.723*▼	-0.696*▼	-0.395*▼	**-0.562***▼	0.214*▲	-0.112	0.191	-0.026	0.389*▼	0.050	0.135
MMD-G	*INST − DOM*	-0.432*▼	-0.568*▼	-0.398*▼	-0.421*▼	0.089*▼	0.015	0.147	0.013	0.194*▼	0.184*▲	0.094
	CTD − DOM	**-0.727***▼	-0.699*▼	-0.391*▼	-0.547*▼	0.218*▲	-0.113	0.179	-0.019	0.394*▼	0.047	0.122
	INST − INST	-0.718*▼	**-0.711***▲	**-0.458***▼	**-0.567***▼	**0.255***▲	-0.094	0.186	0.001	**0.426***▲	0.062	0.149
	CTD − CTD	-0.723*▼	-0.696*▼	-0.396*▼	-0.563*▼	0.214*▲	-0.112	0.191	-0.026	0.389*▼	0.050	0.134
MMD-L	*INST − DOM*	-0.192*▼	-0.519*▼	-0.363*▼	-0.387*▼	0.211*▲	-0.056	0.173	0.004	0.275*▼	-0.006	0.161
	CTD − DOM	-0.733*▼	-0.702*▼	-0.390*▼	-0.565*▼	0.221*▲	-0.117	0.187	-0.010	0.398*▼	0.041	0.126
	INST − INST	**-0.735***▼	**-0.716***▲	**-0.456***▼	**-0.630***▲	**0.257***▲	-0.100	0.240	0.011	**0.432***▲	0.072	0.178
	CTD − CTD	-0.734*▼	-0.701*▼	-0.386*▼	-0.575*▼	0.212*▲	-0.106	0.184	-0.004	0.389*▼	0.060	0.121
MMD-E	*INST − DOM*	-0.283*▼	-0.529*▼	-0.384*▼	-0.434*▼	0.165*▼	-0.016	0.166	-0.007	0.244*▼	0.085	0.130
	CTD − DOM	-0.731*▼	-0.702*▼	-0.388*▼	-0.559*▼	0.220*▲	-0.117	0.183	-0.013	0.397*▼	0.042	0.124
	INST − INST	**-0.734***▼	**-0.713***▲	**-0.465***▼	**-0.579***▼	**0.259***▲	-0.095	0.202	0.001	**0.432***▲	0.066	0.159
	CTD − CTD	-0.731*▼	-0.700*▼	-0.387*▼	-0.571*▼	0.213*▲	-0.108	0.182	-0.010	0.389*▼	0.057	0.122
XPOS Distributions												
Bhat.	*DOM − DOM*	**-0.816***	**-0.687***	-0.542*	**-0.668***	**0.234***	-0.116	0.269	-0.012	**0.438***	0.068	0.222
	INST − DOM	0.131*▼	0.143	-0.163	-0.237*▼	0.091*▼	0.164*▲	0.081	0.043	0.087*▼	0.134	0.117
	CTD − DOM	-0.807*▼	-0.674*▼	-0.541*▼	-0.655*▼	**0.234***▬	-0.119	0.260*▲	-0.019	0.437*▼	0.060	0.220
	INST − INST	0.282*▼	-0.056	-0.127	-0.279*▼	-0.090*▼	-0.074	-0.074	0.061	-0.106*▼	**-0.302***▼	-0.056
	CTD − CTD	-0.796*▼	-0.669*▼	**-0.534***▼	**-0.668***▬	0.225*▼	-0.120	**0.264***▲	-0.024	0.422*▼	0.053	0.225
CORAL	*INST − DOM*	0.218*▼	**0.311***▼	0.008	0.163*▼	0.027	**0.190***▲	0.006	0.013	0.005	0.088	0.050
	CTD − DOM	0.165*▼	0.266*▼	0.019	0.102	0.019	0.185*▲	0.028	-0.067	0.006	0.079	0.067
	INST − INST	**0.256***▼	0.131	-0.120	0.150*▼	-0.044	-0.046	-0.037	0.063	-0.063*▼	**-0.380***▲	0.013
	CTD − CTD	0.164*▼	0.087	-0.151	**0.180***▼	-0.009	0.079	0.041	**-0.152***▲	-0.024	-0.142	0.091
CMD	*INST − DOM*	-0.509*▼	-0.555*▼	-0.453*▼	-0.469*▼	0.096*▼	0.029	0.132	0.037	0.217*▼	**0.209***▲	0.101
	CTD − DOM	**-0.759***▼	-0.684*▼	-0.443*▼	-0.611*▼	0.227*▼	-0.087	0.217	-0.009	0.412*▼	0.076	0.168
	INST − INST	-0.719*▼	**-0.714***▲	**-0.550***	-0.597*▼	**0.257***▲	-0.061	0.244	-0.009	**0.435***▼	0.092	0.226
	CTD − CTD	-0.756*▼	-0.684*▼	-0.449*▼	**-0.620***▼	0.223*▼	-0.085	0.222	-0.014	0.407*▼	0.080	0.175
MMD-G	*INST − DOM*	-0.428*▼	-0.553*▼	-0.445*▼	-0.481*▼	0.138*▼	0.025	0.153	0.009	0.244*▼	**0.199***▲	0.119
	CTD − DOM	**-0.759***▼	-0.684*▼	-0.441*▼	-0.612*▼	0.227*▼	-0.087	0.216	-0.010	0.412*▼	0.076	0.167
	INST − INST	-0.744*▼	**-0.713***▼	**-0.517***▼	-0.613*▼	**0.260***▲	-0.061	0.225	-0.007	**0.440***▲	0.094	0.192
	CTD − CTD	-0.756*▼	-0.684*▼	-0.447*▼	**-0.622***▼	0.223*▼	-0.085	0.224	-0.013	0.407*▼	0.080	0.177
MMD-L	*INST − DOM*	-0.096*▼	-0.470*▼	-0.364*▼	-0.398*▼	0.194*▼	0.005	0.179	0.005	0.245*▼	0.075	0.182
	CTD − DOM	**-0.764***▼	-0.688*▲	-0.437*▼	-0.618*▼	0.231*▼	-0.091	0.222	-0.007	0.417*▼	0.072	0.165
	INST − INST	-0.746*▼	**-0.712***▲	**-0.480***▼	**-0.663***▼	**0.266***	-0.072	**0.269***▲	-0.018	**0.441***▲	0.108	0.217
	CTD − CTD	-0.768*▼	-0.685*▼	-0.431*▼	-0.630*▼	0.224*▼	-0.082	0.228	0.003	0.411*▼	0.087	0.168
MMD-E	*INST − DOM*	-0.203*▼	-0.502*▼	-0.412*▼	-0.472*▼	0.181*▼	0.019	0.168	-0.002	0.249*▼	0.134	0.152
	CTD − DOM	-0.763*▼	-0.687*▬	-0.440*▼	-0.616*▼	0.229*▼	-0.089	0.222	-0.010	0.415*▼	0.073	0.168
	INST − INST	-0.750*▼	**-0.713***▲	**-0.519***▼	-0.621*▼	**0.265***▲	-0.062	0.246	-0.009	**0.444***	0.102	0.206
	CTD − CTD	**-0.765***▼	-0.685*▼	-0.430*▼	**-0.626***▼	0.224*▼	-0.083	0.219	-0.002	0.410*▼	0.084	0.162

to their similarities. The observations from Table 6.7 echo those from the previous tables, with one notable difference being that the Laplacian variant of MMD demonstrates marginally better performance than that of the Gaussian kernel. As such, for both UPOS and XPOS, we select the following measures and aggregation methods for use in our task

selection experiments further in this thesis: Bhattacharyya distance ($DOM - DOM$), CMD ($INST - INST$), and MMD-L ($INST - INST$).

In response to RQ5, the analysis demonstrates that instance-level comparisons are effective—especially for NE distributions in Yahoo and TREC-IS—with LFDs, supporting our initial hypothesis that instance-based comparisons are more effective when the dimensions of distributions are smaller. Bhattacharyya distance emerges as a consistently strong performer in zero-shot settings across various LFD types, underscoring its effectiveness in capturing domain divergence. The Gaussian and Laplacian variants of MMD shows promise, particularly in $INST - INST$ comparisons. The inconsistency of CORAL across different distributions and settings limits its applicability for LFDs. CMD's success in instance-level comparisons for specific datasets also highlights its potential. Overall, Bhattacharyya and MMD-G stand out as particularly effective across multiple distribution types and aggregation levels. The analysis underscores the importance of tailored approaches to divergence measurement and aggregation for different distributions.

6.6 Topic Frequency Distributions Evaluation

Table 6.8: Spearman's ρ correlations between domain (DOM) representation divergence and intermediate-to-target F_1 scores. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of measures tested ($p < 0.006$). The **bold** values highlight the strongest correlations for each dataset and setting. Symbols ▲ and ▼ compare values to their respective Cos. baselines (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

	Zero-shot				Few-shot				Limited		
Measure	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Cos.	-0.592*	-0.796*	-0.642*	-0.439*	0.289*	-0.021	0.037	-0.140	0.201*	0.118	-0.083
L_1	-0.614*▲	-0.794*▲	-0.667*▲	-0.407*▼	**0.297***▲	-0.044	0.075	-0.042	0.215*▲	0.073	-0.044
L_2	-0.615*▲	-0.723*▼	**-0.735***▲	**-0.453***▲	0.295*▲	0.018	0.179	-0.019	0.223*▲	**0.209***▲	0.155
JS	-0.630*▲	**-0.816***▲	-0.677*▲	-0.384*▼	0.290*▲	-0.035	0.071	-0.061	0.239*▲	0.094	-0.050
Rényi	**-0.640***▲	-0.803*▲	-0.685*▲	-0.326*▼	0.270*▼	-0.093	0.071	0.004	**0.289***▲	0.045	-0.035
Bhat.	-0.635*▲	-0.814*▲	-0.684*▲	-0.377*▼	0.294*▲	-0.035	0.075	-0.056	0.250*▲	0.099	-0.050
Wass-1	-0.615*▲	-0.723*▼	**-0.735***▲	**-0.453***▲	0.295*▲	0.018	0.179	-0.019	0.223*▲	**0.209***▲	0.155
Wass-2	-0.615*▲	-0.723*▼	**-0.735***▲	**-0.453***▲	0.295*▲	0.018	0.179	-0.019	0.223*▲	**0.209***▲	0.155

The focus of this experiment is on evaluating the effectiveness of Topic Frequency Distributions (KFDs) (at the domain level) in predicting transfer learning performance. We assess the correlations between the divergence of these distributions and relative transfer gain, using various divergence measures. The divergence measures include Cosine, L_1, L_2, Jensen-Shannon (JS), Rényi, Bhattacharyya (Bhat.), and Wasserstein-1 and -2. Table 6.8 presents Spearman's ρ correlations between the divergence of domain-level KFD representations and F_1 scores for various datasets. We observe the following:

- **l_p-norm measures exhibit superior performance over Cosine**. The L_2 distance, in particular, shows a marked improvement over the Cosine distance, especially in terms of its correlation strength with F_1 scores.
- **Information-theoretic measures are very effective**. JS and Rényi divergences demonstrate strong correlations, particularly for the Amazon and Yelp datasets.
- **Wasserstein-based measures are both effective and consistent across all settings**. Both Wasserstein-1 and -2 maintain consistent good performance across various datasets. Their stable correlation strengths position them as reliable measures for evaluating KFDs in subsequent experiments.

In response to RQ6, the analysis reveals that Wasserstein measures stand out for their consistent performance in capturing domain divergence in KFDs, making them particularly suitable for use in task selection experiments within transfer learning scenarios. Additionally, the L_2 distance emerges as a more effective measure compared to Cosine distance, showing substantial improvements in correlation strength. The strong performance of information-theoretic measures, particularly JS and Rényi, further underscores their potential utility for KFDs.

6.7 Representation Configuration Summary

This section provides a summary of the most effective configurations for the distributional representations discussed in this chapter. It details the aggregation methods that correlated most strongly with relative transfer gain for each type of distribution, along with specific representation configurations. These identified configurations will be used in the feature engineering stage for the intermediate task selection process discussed in Chapter 8.

Term Distributions. In TF (Term Frequency) distributions, the DOM-DOM aggregation method paired with Cosine and Rényi divergences emerged as the top performer. The optimal vocabulary configuration for this model was TRM=TF, DC=$\mathcal{D}_T$. In the case of TF-IDF distributions, the same aggregation method and divergence measures were applied. However, here, the vocabulary was derived from the target domain, with terms ranked by TF-IDF, i.e. TRM=TF-IDF, DC=$\mathcal{D}_T$.

Linguistic Feature Distributions. For Linguistic Feature Distributions involving Named Entities (NE), the $DOM-DOM$ aggregation method with Bhattacharyya distance yielded the best results. Additionally, NE distributions also performed well under both CMD and MMD (Gaussian kernel) using the $INST-INST$ aggregation method. U/XPOS distributions followed a similar pattern, with $DOM-DOM$ and Bhattacharyya distance as the

leading method, with competitive results from CMD and MMD (Laplacian kernel) under $INST - INST$ aggregation.

Topic Frequency Distributions. The analysis of Topic Frequency Distributions showed that the Wasserstein-1 divergence measure was most effective in yielding strong performance results.

In the following section, we will conclude this chapter with a discussion of our findings on distributional representations and how these findings contribute to our thesis, offering a comprehensive summary and reflecting on the implications of these results.

6.8 Conclusions

In this section, we focus on the development and evaluation of distributional representations as part of our broader framework for estimating transferability in natural language processing. Our approach was methodical, dividing the process into three distinct phases: constructing distributional representations, evaluating their effectiveness, and refining these representations to improve their correlation with transfer gains.

First, in Section 6.2, we described our methodology to create distributional representations from domain-specific corpora and to evaluate these representations through a series of correlation analyses to assess how the divergence of these representations correlated with transfer gains in intermediate-to-target models.

Second, in Section 6.3, we defined the research questions that we investigated in this chapter. We divided our approach into six individual experiments. The first three examined the evaluation of Term Distributions, questions four and five examined the evaluation of Linguistic Feature Distributions, and the final question examined Topic Frequency Distributions. The first set of experiments on Term Distributions involved (1) establishing a baseline with a target-focused vocabulary, selecting terms based on their frequency in the target corpus; (2) aiming to improve this baseline by evaluating divergence at different abstraction levels, specifically at centroid and domain levels; (3) adjusting our vocabulary construction methods for term distributions to assess their influence on correlation outcomes. In the case of Linguistic Feature Distributions, we started by setting a baseline for Named Entity, Linguistic Dependency, and Part-of-Speech distributions before concluding with an evaluation applying our aggregation strategy. The last set of experiments centred on training and evaluating topic models to generate and assess Topic Frequency Distributions.

Through experiments detailed in Sections 6.4, 6.5, and 6.6, our findings revealed that most distributional representations showed a strong link between domain divergence and transfer learning performance, particularly in zero-shot contexts, while others demon-

strated varied effectiveness in more challenging experimental settings. Our term distribution studies highlighted the importance of approaches to vocabulary construction, particularly in few-shot and limited contexts. Furthermore, we found that, for these distributions, varying the level of abstraction for divergence calculation (centroid and domain) did not significantly improve performance over our baselines. However, certain statistical moments-based measures such as Maximum Mean Discrepancy were effective for lower-dimensional distributions such as Linguistic Feature Distributions. Topic Frequency Distributions, particularly with Wasserstein and information-theoretic measures, showed promise in capturing domain divergence effectively.

The findings from this chapter directly contribute to answering our first thesis research question (RQ1 in Section 1.3), which focuses on the construction and evaluation of effective domain representations. Our experiments demonstrate that distributional representations—namely Term Distributions, Linguistic Feature Distributions, and Topic Frequency Distributions—can effectively capture domain characteristics and relationships. The strong correlations between representation divergence and transfer performance metrics in zero-shot settings support our expectation that advanced domain representations and precise divergence estimations can significantly improve the accuracy of transferability estimation. However, the varied effectiveness of these representations in more challenging experimental settings highlights the need for further investigation into embedding-based approaches, which we address in the following chapter.

In the forthcoming experimental chapter, we examine if embedding representations similarly correlate with transfer learning performance. In particular, we explore BERT embeddings (both general-purpose/static and contextual embeddings), weighting embeddings by frequency-based information contained within our term distributions, and finally, combining theoretical approaches from both distributional and embedding-based approaches to produce distributions of contextually-aggregated term vectors and their role in predicting the success of transfer learning. This exploration aims to further substantiate our hypothesis that creating and comparing effective domain representations enhances the identification of similar tasks conducive to transfer.

Chapter 7

Embedding Representations

7.1 Introduction

In the previous chapter, we performed a systematic analysis on quantifying domain divergence using distributional representations. The findings showed that most representations had strong correlations in zero-shot settings. Information-theoretic measures were particularly effective, followed by geometric measures. The analysis highlighted the effectiveness of domain-level comparisons across different types of distributions, with instance- and centroid-level comparisons also yielding similarly strong correlations for lower-dimensional Linguistic Feature Distributions (LFD).

This chapter investigates embeddings in the context of estimating transferability in natural language processing. Embeddings are a form of vector representation that transform words or phrases from a vocabulary into vectors of real numbers. Unlike probability distributions, embeddings do not sum up to one or necessarily fall within a specific range. Instead, they occupy a multi-dimensional space where each dimension represents a latent feature of the word, capturing semantic and syntactic properties. The origins of embeddings can be traced back to the vector space model developed by Salton et al. [93], which implemented distributional data in a sparse, high-dimensional vector space. Bengio et al. [4] introduced *Neural Probabilistic Language Models* to address the limitations of high-dimensional word representations. This methodology streamlined semantic and syntactic analysis through distributed representations, reducing complexity without compromising the depth of linguistic relationships. This groundwork was expanded with the introduction of *Word2Vec* by Mikolov et al. [61], using Continuous Bag-of-Words (CBOW) and Skip-Gram models for embedding generation. While Word2Vec efficiently handled large datasets and captured diverse word relationships, their *static* word embeddings fell short in addressing word polysemy. In contrast, BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al. [26], using the Transformer [108] architecture, introduces dynamic, *contextual* embeddings. These embeddings are sensitive to the surround-

ing words, allowing for a more accurate representation of words used in different linguistic contexts.

The objective of this chapter is to examine embedding-based representations and their impact on understanding domain similarities and divergences in natural language processing. Embeddings, with their rich, multidimensional representation of language, are expected to provide stronger correlations compared to the frequency-based analysis of distributional representations. These representations provide a comprehensive view of semantic and syntactic relationships, enabling a more detailed measurement of linguistic proximity between domains. We begin with exploring the capabilities of BERT embeddings in both static and contextual forms, examining their correlation with transfer learning performance. Static embeddings provide context-independent word representations, offering efficiency in computation. In contrast, contextual embeddings from BERT offer richer semantic information by encoding the word's surrounding context within the embedding. Similarly to the previous chapter, we conduct an aggregation analysis to determine how suitable instance- and centroid-based divergence computation is for embeddings.

Building on these foundations, we introduce Probability-Weighted Embeddings (PWE), an approach that integrates frequency-based information from term distributions with embeddings. By weighting term vectors based on their frequency within a corpus, PWE aims to enhance the representation by combining the significance of terms with their semantic meanings. This method is expected to yield stronger correlations between divergence and transfer gain, as evidenced by the strong performance of term distribution-based representations in the previous chapter.

Lastly, we present Distributive Contextual Embeddings (DCE), a technique that merges the concepts of distributional and embedding representations. DCE involves aggregating contextual embeddings of terms across documents, creating a domain-level representation of our vocabularies. This method allows for a closer comparison of the semantic context variations of individual terms, complementing the broader document and domain vector comparisons. In particular, we use measures that describe the characteristics of distributions—entropy- and statistical moments-based measures—to describe the diversity of term distances.

Each of these methods—BERT embeddings, PWE, and DCE—contributes uniquely to our understanding of domain divergence using embedding representations. While BERT embeddings provide a baseline of contextual and static representations, PWE enhances these embeddings with term frequency data. DCE, on the other hand, represents a more advanced integration of distributional and contextual information, adding another dimension to our analysis.

7.2 Methodology

7.2.1 Evaluation Methodology

In this chapter, we adopt the same methodological framework for evaluation as detailed in the previous chapter, particularly focusing on Spearman's correlation analysis and aggregation strategies. Our hypothesis for this set of experiments remains consistent, positing an inverse relationship between domain divergence and transfer learning effectiveness.

However, specific to this chapter, we introduce a few adaptations to our methodology. First, we introduce a competitive baseline comparison, Sentence-BERT (SEmb) [85] embeddings, which have been shown to perform well for transferability estimation [72]. Sentence-BERT embeddings are derived from the BERT model fine-tuned on a large corpus of sentence pairs and are designed to capture sentence-level semantics. By including this baseline, we aim to compare the performance of our embedding representations for transferability estimation against a well-established and state-of-the-art embedding approach.

In the previous chapter, we focused on Distributional Representations (TD, LFD, KFD), where each representation captured a distinct characteristic of the data, and the dimensions and properties varied significantly. In contrast, this chapter primarily focuses on BERT embeddings and their weighted equivalents, which are characteristically similar. Given the inherent uniformity of these embeddings and computational considerations, our aggregation strategies (Section 6.2.2) are slightly modified. We perform aggregations only once, as the intrinsic properties across different types of embeddings in this chapter do not exhibit the same degree of variability as the representations in the previous chapter. Furthermore, we prioritise methods that are most efficient for embeddings, such as domain-to-domain ($DOM - DOM$), instance-to-domain ($INST - DOM$), and centroid-to-domain ($CTD - DOM$) comparisons. Similarly to Term Distributions (Section 6.4), we omit instance-instance ($INST - INST$) calculations due to their computational expense.

7.2.2 BERT Embeddings

In this work, we use two types of BERT embeddings, static and contextual, each with benefits and drawbacks:

- **Static BERT Embeddings (BERT (S))**. These are derived from BERT's word embedding layer and offer general representations of words based on the model's training data. While they do not capture the detailed context of dynamic embeddings, static embeddings are computationally less demanding, as they avoid a full forward pass through the neural network.

- **Contextual BERT Embeddings (BERT (C))**. These embeddings are generated by processing text data through BERT in a full forward pass, specifically extracting them from the second-to-last layer. This layer selection is based on guidance from the original authors [26], who indicate that it provides a more detailed and context-rich representation than the final output layer in Named Entity Recognition tasks.

BERT uses *WordPiece* for subword segmentation, which helps handle morphological variations and rare terms. This results in words being divided into several subwords, requiring a strategy to reconstruct complete term representations. We address this by averaging the embeddings of these subword parts. This post-processing step ensures that each term, regardless of its segmentation, is represented by a unified vector, thereby maintaining the semantic information from its divided components.

7.2.3 Probability-Weighted Embeddings

Probability-Weighted Embeddings (PWE) combine term frequency statistics with the semantic context captured by word embeddings, integrating distributional and embedding-based information in a single representation. PWE assign weights to terms in a document based on their prevalence within a target domain vocabulary. Terms that do not exist in the target domain vocabulary are unchanged. The weighting scheme is designed to amplify the influence of terms that are characteristic of the target domain, facilitating a more meaningful comparison of embeddings representing similar contexts based on term frequency.

Consider a domain $\mathcal{D}$ and its document set $\mathbf{D}_{\mathcal{D}}$. Let $\mathbf{V}_{\mathcal{D}_T}$ denote the target domain-specific vocabulary. For any document $d \in \mathbf{D}_{\mathcal{D}}$, we assign weights to those terms that exist in $\mathbf{V}_{\mathcal{D}_T}$ based on their occurrence in $\mathcal{D}$. Specifically, a term $t_{i,d}$ is assigned a weight calculated by the formula:

$$w(t_{i,d}) = 1 + P(t_{i,d})$$

where $P(t_{i,d})$ is the document-level probability of the term $t_{i,d}$ if $t_{i,d} \in \mathbf{V}_{\mathcal{D}_T}$, and $P(t_{i,d}) = 0$ otherwise. This approach ensures that only terms within the target domain-specific vocabulary are used to weight the embedding. An increment of 1 is applied in order to: (1) ensure the embedding is unchanged in the event that $P(t_{i,d}) = 0$; or (2) in the event that $P(t_{i,d}) \neq 0$, ensure that the information encoded within the embedding is not diminished for low-frequency terms.

For domain-level term weighting, where we consider the frequency of a term across the entire document set $\mathbf{D}_{\mathcal{D}}$, the weight of term $t_{i,d}$ is given by:

$$w(t_{i,d}) = 1 + P(t_{i,\mathbf{D}_{\mathcal{D}}})$$

where $P(t_{i,\mathbf{D}_\mathcal{D}})$ is the probability of term $t_{i,d}$ across $\mathbf{D}_\mathcal{D}$ if $t_{i,d} \in \mathbf{V}_{\mathcal{D}_T}$, and $P(t_{i,\mathbf{D}_\mathcal{D}}) = 0$ otherwise.

7.2.4 Distributive Contextual Embeddings

Distributive Contextual Embeddings (DCE) capture the contextual variation of terms across different domains, with a specific focus on terms present in a target domain vocabulary. This approach compares embeddings of terms that are included in the target domain vocabulary by averaging their embeddings from all documents within each domain where these terms occur. The resulting embeddings represent each term's average contextual representation within each domain, facilitating analyses of contextual divergence between domains on a term-to-term basis.

Mathematically, for terms in the target domain vocabulary $\mathbf{V}_{\mathcal{D}_T}$, the representation of a term t_i is determined by averaging its embeddings across various documents within the domain. The mathematical representation is expressed as follows:

$$\vec{\bar{t}_i} = \frac{1}{N_{t_i}} \sum_{d \in \mathbf{D}_\mathcal{D}} \vec{t_{i,d}} \quad \forall t_i \in \mathbf{V}_{\mathcal{D}_T}$$

where $\vec{t_{i,d}}$ is the embedding of term t_i in document d from the set of documents belonging to a particular domain $\mathbf{D}_\mathcal{D}$, N_{t_i} is the frequency of t_i in $\mathbf{D}_\mathcal{D}$, and $\mathbf{V}_{\mathcal{D}_T}$ is the target domain vocabulary.

The DCE for a domain comprises an aggregation of these averaged term embeddings, creating a composite representation that reflects the overall contextual usage of terms across both document sets. Our analysis of DCEs involves specific steps, each designed to provide insights into the characteristics and differences of domain representations. These steps are:

1. **Baseline Comparison**: This analysis computes the average pairwise distances between vectors of term features in different domains, constrained by a target vocabulary, $\mathbf{V}_{\mathcal{D}_T}$. It provides a foundational understanding of domain relatedness from a term-centric perspective.

2. **Histogram Analysis**: This method segments term-to-term distances into bins for a structured view of distance distribution. It addresses the limitations of averaging instance-level comparisons by categorising distances, allowing for a granular analysis of proximity of like terms across domains. The analysis includes the comparison of normalised frequency distributions, calculation of statistical moments, and incorporation of diversity metrics like Entropy and Simpson's Index. These provide a comprehensive statistical profile of the distance distributions, offering insights into the variety and characteristics of these distributions.

3. **Distributional Distance Evaluation**: This analysis uses the Wasserstein metric to evaluate the effort required to transform the distribution of term-to-term distances in one domain to match another. The rationale behind this approach lies in conceptualising domain similarity in terms of the cost of transforming one distribution into another.

   A distribution of zeros is chosen as the target for comparison. This zero distribution represents an ideal scenario where all term vectors in a domain are identical, signifying complete domain equivalence. The Wasserstein metric then quantifies the "effort" or "cost" needed to transform the actual distribution of term or feature distances into this ideal zero distribution.

   The lower the Wasserstein distance, the smaller the transformation effort required, indicating a higher degree of similarity between the domains. In essence, this metric provides a quantitative measure of how closely the terms or features in one domain resemble those in another. By using a zero distribution as a reference, we can objectively assess the extent to which the actual term distributions deviate from this ideal state of complete similarity, thereby offering a theoretically grounded approach to quantify domain relatedness.

7.3 Research Questions

RQ1: What is the relative effectiveness of geometric divergence measures in capturing the correlation between embedding-based representations and transfer learning performance, as indicated by Spearman's correlation with model performance scores?

This research question examines the effectiveness of geometric divergence measures in correlating embedding-based representations with transfer learning performance. The focus is to determine how these measures, when applied to embeddings, relate to the success of transfer learning models, using Spearman's correlation with model performance scores as an indicator.

RQ2: How do different aggregation methods (instance-domain, centroid-domain, and centroid-to-centroid) influence the representational quality and correlation of embedding-based representations with transfer learning performance compared to baseline domain-aggregated methods?

This question explores the impact of various aggregation methods on the efficacy of embedding-based representations in transfer learning contexts. By comparing instance-

domain, centroid-domain, and centroid-to-centroid methods against the baseline domain-aggregated approach, we aim to understand how different aggregation strategies affect the representational quality and their correlation with relative transfer gain.

RQ3: How do embeddings, weighted by term distribution-based probabilities compare to both static and contextual embeddings, in terms of their correlation with model performance scores?

RQ3 investigates the performance of embeddings weighted by term distribution-based probabilities. The goal is to assess how these probability-weighted embeddings compare with standard static and contextual embeddings in terms of their correlation with transfer learning model performance scores.

RQ4: How do Distributive Contextual Embeddings perform in predicting intermediate tasks for transfer, and how do they compare to both BERT embeddings and probability-weighted embeddings?

The focus of this research question is on evaluating the performance of Distributive Contextual Embeddings (DCEs) in predicting suitable intermediate tasks for transfer learning. This involves comparing the effectiveness of DCEs against BERT embeddings and probability-weighted embeddings, examining their respective capabilities in aiding the selection of intermediate tasks based on their correlation with relative transfer gain.

7.4 BERT Embeddings Evaluation

In this experiment, we analyse the effectiveness of different embedding representations, specifically Sentence Embeddings (SEmb) and BERT embeddings (both static and contextual variants), in predicting transfer learning performance. The analysis compares these embeddings across various datasets—Amazon (AM), Yelp (YL), Yahoo (YH), and TREC-IS (IS)—using geometric divergence measures like Cosine, L_1, L_2 distances, and Wasserstein-1 and -2. The aim is to discern which type of embedding and divergence measure most accurately reflects the correlation with transfer learning success, as measured by Spearman's correlation with model performance scores.

Table 7.1 presents Spearman's ρ correlations for each divergence measure against their relative transfer gain (intermediate-to-target F_1 scores) across the datasets Amazon (AM), Yelp (YL), Yahoo (YH), and TREC-IS (IS) and experimental settings Zero-shot, Few-shot and Limited. The format of this table is similar to evaluations in Chapter 6, with significant correlations marked with an asterisk and the p-value adjusted for the number of representation combinations within each measure ($p < \frac{0.05}{3} = 0.017$). Measures include BERT (S) and (C), which denote the static and contextual variants of BERT, respectively, and

Table 7.1: Spearman's ρ correlations between domain (DOM) representation divergence and intermediate-to-target F_1 scores are presented. "Repr." specifies the representation type: Sentence Embedding (SEmb) or BERT (with "S" for static and "S" for contextual variants). Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of representation combinations within each measure ($p < 0.017$). The **bold** values highlight the strongest correlations within each measure for each dataset and setting, best overall scores are underlined. Symbols ▲ and ▼ compare values to the SEmb baseline (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Repr.	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Cos.	SEmb	-0.873*	**-0.858***	-0.726*	-0.698*	0.220*	0.014	0.361*	0.060	0.455*	0.215*	0.402*
	BERT (S)	-0.811*▼	-0.825*▼	-0.530*▼	-0.594*▼	**0.309***▲	-0.024	0.263*▼	0.017	**0.503***▲	0.157*▼	0.261*▼
	BERT (C)	**-0.904***▲	-0.787*▼	**-0.787***▲	**-0.766***▲	0.272*▲	-0.063	**0.435***▲	0.007	0.489*▲	0.105	**0.497***▲
L_1	SEmb	-0.882*	**-0.854***	-0.640*	-0.642*	0.242*	0.031	0.421*	0.056	0.472*	**0.216***	0.407*
	BERT (S)	-0.810*▼	-0.821*▼	-0.532*▼	-0.594*▼	**0.308***▲	-0.027	0.266*▼	0.019	**0.501***▲	0.149	0.268*▼
	BERT (C)	**-0.904***▲	-0.790*▼	**-0.782***▲	**-0.761***▲	0.270*▲	-0.059	**0.440***▲	0.011	0.486*▲	0.109	**0.499***▲
L_2	SEmb	-0.884*	**-0.855***	-0.641*	-0.646*	0.243*	0.026	0.414*	0.062	0.474*	0.212*	0.400*
	BERT (S)	-0.811*▼	-0.821*▼	-0.530*▼	-0.585*▼	**0.309***▲	-0.026	0.263*▼	0.013	0.502*▲	0.149	0.262*▼
	BERT (C)	**-0.904***▲	-0.787*▼	**-0.774***▲	**-0.764***▲	0.270*▲	-0.060	**0.435***▲	0.010	**0.486***▲	0.107	**0.496***▲
Wass-1	SEmb	-0.884*	**-0.855***	-0.640*	-0.646*	0.243*	0.026	0.414*	0.062	0.474*	0.213*	0.400*
	BERT (S)	-0.811*▼	-0.821*▼	-0.530*▼	-0.585*▼	**0.309***▲	-0.026	0.263*▼	0.013	0.502*▲	0.149	0.262*▼
	BERT (C)	**-0.904***▲	-0.787*▼	**-0.774***▲	**-0.764***▲	0.270*▼	-0.060	**0.435***▲	0.010	**0.486***▲	0.107	**0.496***▲
Wass-2	SEmb	-0.364*	-0.551*	-0.169	-0.598*	-0.031	**-0.159***	0.092	0.038	0.110*	0.136	0.112
	BERT (S)	-0.811*▲	**-0.821***▲	-0.530*▲	-0.585*▼	**0.309***▲	-0.025	0.263*▲	0.013	**0.502***▲	0.149	0.262*▲
	BERT (C)	**-0.872***▲	-0.785*▲	**-0.711***▲	**-0.764***▲	0.236*▲	-0.063	**0.406***▲	0.010	0.471*▲	0.099	**0.458***▲

our Sentence-BERT (SEmb) baseline. Symbols ▲ and ▼ provide a comparative analysis against the baseline. Values in bold represent the strongest correlations within each measure, and those underlined signify the strongest overall correlations per dataset/setting. Key observations include:

- **Sentence Embeddings (SEmb) are competitive as a baseline**. SEmb exhibits strong performance, closely aligning with or even exceeding BERT (C) in specific scenarios, notably in Yelp under few-shot conditions using Wasserstein-2. This indicates SEmb's potential as an effective and cost-efficient baseline for embedding representations.

- **Contextual BERT embeddings are very strong in zero-shot settings**. BERT (C) consistently achieves the strongest correlations in zero-shot settings across datasets. Cosine distance is the most effective measure for BERT (C), outperforming other measures and variants in most (3 out of 4) datasets.

- **Static embeddings compete with their contextual counterparts in certain contexts**. In some cases, like Amazon-Few, static BERT embeddings outperform the contextual variant. This suggests that simpler, static embeddings can capture relevant domain information efficiently, offering a less computationally expensive alternative. Indeed, the performance of static embeddings surpass their contextual variants 13 times amongst all correlation values.

- **Few-shot and Limited settings are very hard to predict**. Performance varies significantly in few-shot and limited settings, indicating the challenge of predicting transfer gain in these scenarios. For instance, Yelp only shows one significant correlation in few-shot settings, and TREC-IS has no significant results. Yelp also has limited success in the limited setting, reporting significance in only 5 out of 15 observations.

Addressing RQ1, this analysis shows that while Sentence Embeddings are effective in some cases, Cosine distance combined with contextual BERT embeddings (BERT (C)) is generally the most effective for capturing correlations with transfer learning performance, especially in zero-shot settings. This suggests the importance of context in BERT (C) for domain-specific tasks. However, static BERT embeddings also show potential in certain few-shot scenarios, serving as a more resource-efficient alternative. Using Cosine distance, BERT (C) often outperforms both static embeddings and the baseline. Thus, we select BERT (C) with Cosine distance for our subsequent aggregation analysis.

7.4.1 Aggregation Analysis

Table 7.2: Spearman's ρ correlations between BERT (C) divergence at different levels of aggregation and intermediate-to-target F_1 scores are presented. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). *INST*, *CTD*, and *DOM* represent instance-, centroid-, and domain-level aggregation. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of aggregation measures tested ($p < 0.017$). The **bold** values highlight the strongest correlations within each measure for each dataset and setting, best overall scores are underlined. Symbols ▲ and ▼ compare values to the BERT (C) (*DOM-DOM*) baseline (using Cosine distance) from the previous experiment within each measure, representing higher or lower correlation magnitudes, respectively.

		Zero-shot				Few-shot				Limited		
Measure	Agg ($\mathcal{D}_I/\mathcal{D}_T$)	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Cos.	$DOM - DOM$	**-0.904***	**-0.787***	-0.787*	-0.766*	0.272*	-0.063	0.435*	0.007	**0.489***	0.105	0.497*
Cos.	$INST - DOM$	-0.729*▼	-0.709*▼	-0.540*▼	-0.514*▼	**0.274***▲	-0.042	0.379*▼	0.074	0.459*▼	0.103	0.460*▼
	$CTD - DOM$	-0.903*▼	**-0.787***▬	-0.792*▲	-0.766*▬	0.272*▬	-0.062	**0.441***▲	0.006	0.488*▼	0.106	**0.506***▲
	$CTD - CTD$	-0.902*▼	-0.785*▼	**-0.798***▲	**-0.767***▲	0.271*▼	-0.065	0.437*▲	0.010	0.487*▼	0.103	0.503*▲
CORAL	$INST - DOM$	0.231*▼	**0.239***▼	-0.065	0.094	0.038	**0.168***▲	0.055	-0.005	0.008	0.087	0.031
	$CTD - DOM$	0.228*▼	0.137	-0.007	-0.050	0.035	0.155*▲	0.027	-0.022	0.004	0.076	0.005
	$CTD - CTD$	**0.240***▼	0.119	-0.235	0.001	-0.031	0.128	0.197	-0.000	**-0.090***▼	-0.030	0.210
CMD	$INST - DOM$	-0.047	-0.401*▼	-0.244	-0.183*▼	0.146*▼	-0.032	0.085	-0.017	0.182*▼	0.080	0.159
	$CTD - DOM$	**-0.903***▼	**-0.787***▬	**-0.772***▼	**-0.764***▼	**0.270***▼	-0.059	0.439*▲	0.010	**0.485***▼	0.109	**0.500***▲
	$CTD - CTD$	**-0.903***▼	-0.786*▼	-0.771*▼	-0.761*▼	0.269*▼	-0.060	**0.441***▲	0.014	**0.485***▼	0.108	**0.500***▲
MMD (G)	$INST - DOM$	-0.064*▼	-0.052	0.131	0.028	0.027	-0.001	**-0.315***▼	0.065	-0.058*▼	0.160*▲	**-0.301***▼
	$CTD - DOM$	**-0.580***▼	-0.731*▼	-0.150	**-0.665***▼	0.039	**-0.215***▲	-0.015	0.003	0.187*▼	-0.034	-0.012
	$CTD - CTD$	-0.574*▼	**-0.750***▼	-0.082	-0.522*▼	**0.160***▼	0.057	0.191	-0.001	**0.296***▼	**0.295***▲	0.151
MMD (L)	$INST - DOM$	0.099*▼	-0.045	-0.013	-0.099	-0.097*▼	-0.068	0.071	-0.010	-0.119*▼	0.176*▼	0.118
	$CTD - DOM$	-0.118*▼	-0.471*▼	-0.146	-0.122	-0.091*▼	0.380*▲	0.073	-0.142	-0.145*▼	**0.563***▲	0.077
	$CTD - CTD$	**-0.464***▼	**-0.505***▼	0.062	-0.110	**0.199***▼	**0.382***▲	-0.075	-0.006	**0.279***▼	0.528*▲	-0.145
MMD (E)	$INST - DOM$	-0.867*▼	-0.776*▼	**-0.779***▼	-0.751*▼	**0.271***▼	-0.074	0.445*▲	0.020	0.478*▼	0.084	**0.524***▲
	$CTD - DOM$	-0.906*▲	**-0.789***▲	-0.768*▼	**-0.765***▼	0.269*▼	-0.060	0.438*▼	0.012	0.485*▼	0.108	0.495*
	$CTD - CTD$	**-0.907***▲	**-0.789***▲	-0.770*▼	-0.760*▼	**0.271***▼	-0.058	**0.446***▲	0.016	**0.488***▼	0.111	0.502*▲

We extend our analysis to examine how different levels of aggregation—instance-to-domain

($INST - DOM$), centroid-to-domain ($CTD - DOM$), and centroid-to-centroid ($CTD - CTD$)—impact the correlation of contextual BERT embeddings with relative transfer gain. The analysis compares these aggregation methods against the previously established domain-to-domain ($DOM - DOM$) baseline, using the Cosine distance measure, across our datasets and settings. The goal is to determine whether calculating divergence representations created on subsets of data improves correlations with relative transfer gain over our baseline. We anticipate that embeddings, being inherently richer representations compared to frequency-based distributions, will be effective with instance- and centroid-based comparisons.

Table 7.2 presents Spearman's ρ correlations between BERT (C) divergence at various aggregation levels and transfer learning performance, with statistical significance adjusted for the number of measures tested ($p < \frac{0.05}{3}$). Symbols ▲ and ▼ how these aggregations compare to the $DOM - DOM$ baseline from prior experiments, indicating whether alternative aggregation methods provide higher or lower correlation magnitudes. Key observations include:

- **Domain-level aggregation with Cosine distance is preferred for embeddings**. Cosine distance at the domain level consistently shows strong negative correlations across most datasets and settings, underlining its effectiveness in capturing domain characteristics in embeddings.

- **Instance-based comparisons are inconsistent**. Contrasting with their effectiveness in distributional representations (see Sections 6.4.1 and 6.5.1), $INST - DOM$ comparisons show varying degrees of correlation, often lower than the domain-level baseline. However, instance-based calculations with CMD measure stand out, displaying the strongest within-measure correlations in almost all comparisons, highlighting its specific utility.

- **Centroid-based comparisons are effective**. $CTD - CTD$ aggregations show promising results, particularly in zero-shot settings. For instance, MMD-E and Cosine in $CTD - CTD$ aggregation show superior performance to $DOM - DOM$ in Amazon and Yelp datasets in this setting. However, the performance of different kernels fluctuates greatly, with correlations ranging from very strong (> 0.70) in the baseline to non-significant. This inconsistency is more pronounced in few-shot and limited settings.

In response to RQ2, this analysis shows that exploring alternative aggregation methods like $INST - DOM$ and $CTD - CTD$ can lead to improved insights and occasionally superior correlations in specific settings. However, given their variability and higher computational demands, the marginal benefits these methods offer may not justify the extra resources

required. Therefore, the $DOM - DOM$ comparison emerges as a more efficient and practical choice for transfer learning tasks. It balances efficiency and effectiveness, making it suitable for scenarios where computational resources are limited, while still delivering strong performance.

7.5 Probability-Weighted Embeddings Evaluation

In this experiment, we assess the effectiveness of Probability-Weighted Embeddings (PWE), where static BERT embeddings are weighted by term distributions (TF/TF-IDF) at both document- ($\mathbf{D}$) and domain-level ($\mathcal{D}$). This approach is compared against unweighted static and contextual BERT embeddings to understand the impact of term distribution-based weighting on the embeddings' ability to predict transfer learning performance. The analysis aims to determine if and how this weighting enhances the correlation with model performance scores across different datasets. Given the strong performance of term distributions (Section 6.4), we expect that weighting term vectors using frequency-based information will improve performance.

Table 7.3: Spearman's ρ correlations between different representation divergences (using Cosine Distance) and intermediate-to-target F_1 scores are presented. "Repr." specifies the representation type: BERT (with "S" for static and "S" for contextual variants), PWE denotes Probability-Weighted Embeddings with $\mathbf{D}$ and $\mathcal{D}$ representing document- and domain-level probability weighting, respectively. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of weighting schemes tested ($p < 0.013$). The **bold** values highlight the strongest correlations for each dataset and setting, split by static and contextual variants, while best overall values are underlined. Symbols ▲ and ▼ compare values to their corresponding BERT (S/C) baseline (grey rows) representing higher or lower correlation magnitudes, respectively.

	Zero-shot				Few-shot				Limited		
Repr.	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Static Embeddings											
BERT (S)	-0.811*	-0.825*	-0.530*	**-0.594***	**0.309***	-0.024	0.263*	0.017	0.503*	0.157*	0.261*
(TF, $\mathbf{D}$)	-0.836*▲	**-0.843***▲	-0.803*▲	-0.539*▼	**0.309***	-0.007	**0.405***▲	0.071	**0.506***▲	**0.169***▲	**0.448***▲
(TF, $\mathcal{D}$)	-0.832*▲	-0.842*▲	-0.804*▲	-0.546*▼	0.308*▼	-0.007	0.403*▲	0.065	0.504*▲	0.168*▲	0.447*▲
(TF-IDF, $\mathbf{D}$)	**-0.847***▲	-0.842*▲	**-0.809***▲	-0.554*▼	0.307*▼	-0.010	**0.405***▲	0.077	**0.506***▲	0.164*▲	0.447*▲
(TF-IDF, $\mathcal{D}$)	-0.837*▲	-0.842*▲	-0.804*▲	-0.551*▼	0.307*▼	-0.009	0.403*▲	0.068	0.505*▲	0.165*▲	**0.448***▲
Contextual Embeddings											
BERT (C)	**-0.904***	**-0.787***	**-0.787***	**-0.766***	0.272*	-0.063	**0.435***	0.007	0.489*	0.105	**0.497***
(TF, $\mathbf{D}$)	-0.853*▼	-0.782*▼	-0.717*▼	-0.671*▼	**0.286***▲	-0.062	0.370*▼	0.054	**0.498***▲	0.093	0.421*▼
(TF, $\mathcal{D}$)	-0.854*▼	-0.782*▼	-0.722*▼	-0.678*▼	0.285*▲	-0.062	0.366*▼	0.047	**0.498***▲	0.093	0.419*▼
(TF-IDF, $\mathbf{D}$)	-0.852*▼	-0.781*▼	-0.718*▼	-0.705*▼	0.285*▲	-0.063	0.369*▼	0.042	0.497*▲	0.092	0.421*▼
(TF-IDF, $\mathcal{D}$)	-0.854*▼	-0.782*▼	-0.722*▼	-0.704*▼	0.284*▲	-0.062	0.366*▼	0.040	**0.498***▲	0.093	0.419*▼

Table 7.3 presents Spearman's ρ correlations between divergences of unweighted and weighted embeddings and relative transfer gain, using Cosine Distance ($DOM - DOM$) as the divergence measure. The table is divided into static and contextual embedding groups to compare the performance of weighted static and contextual embeddings against their

unweighted counterparts. Different weighting schemes like TF and TF-IDF, derived from document-level ($\mathbf{D}$) and domain-level ($\mathcal{D}$) probabilities, are evaluated. Similar to previous experiments, the threshold for significance is adjusted for the number of weighting schemes tested ($p < \frac{0.05}{4} = 0.013$) and symbols ▲ and ▼ compare weighted embeddings to their static (S) or contextual (C) baselines. Key observations include:

- **Weighted, static embeddings are better than their unweighted counterparts**. Weighted static embeddings generally show improvements over unweighted static BERT (BERT (S)) baselines in zero-shot settings across most datasets. Notably, in 77.7% (28 out of 36) of cases with statistical significance, weighted (static) embeddings outperform unweighted baselines. For example, in the Yahoo dataset's zero-shot scenario, TF-IDF-based document-level weighting substantially exceeds the performance of BERT (S), exceeding the magnitude by 0.277 using TF-IDF, $\mathbf{D}$ weighting.

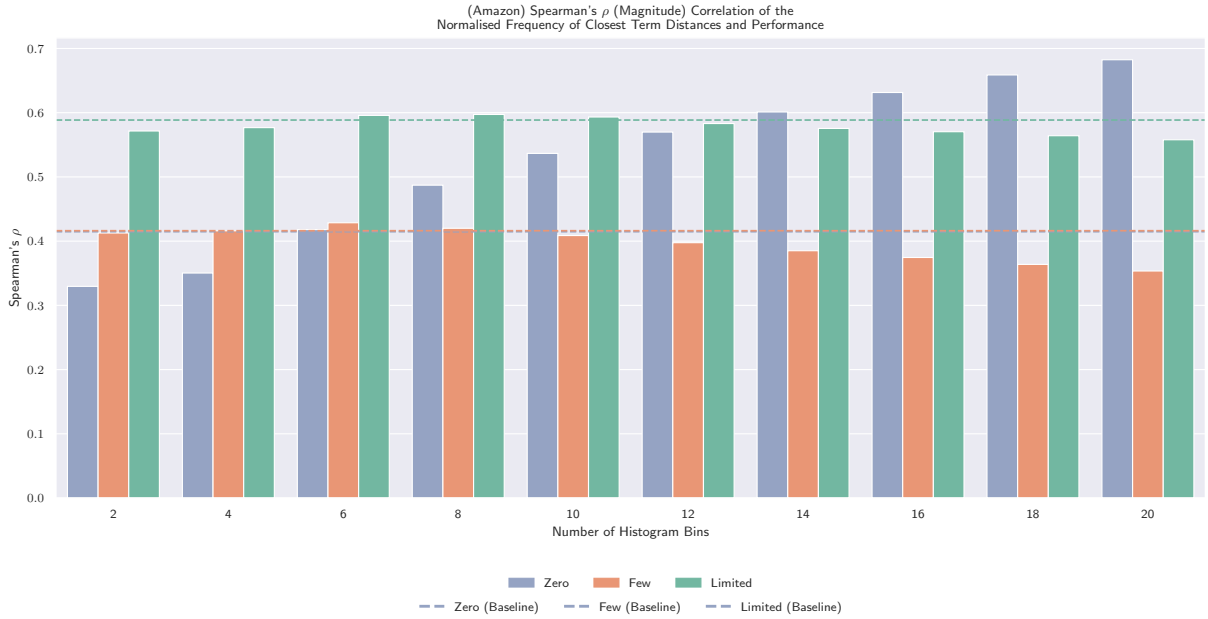
- **Weighting static embeddings with distributional information can rival the performance of unweighted, contextual embeddings**. A remarkable finding is that weighted static embeddings outperform unweighted BERT (C) in 62.5% (5 out of 8) of statistically significant results. This suggests that incorporating distributional representations into static embeddings can produce embeddings of similar quality to the more computationally expensive contextual embeddings, which require a full forward pass over a set of documents.

- **There is limited impact of weighting on contextual embeddings**. Compared to unweighted BERT (C), weighted contextual embeddings demonstrate a slight decrease in zero-shot settings. In few-shot and limited scenarios, though they sometimes perform marginally better, they generally do not add substantial value over the unweighted variant.

- **There are marginal differences between domain- and document-level weighting**. Both document-level ($\mathbf{D}$) and domain-level ($\mathcal{D}$) weighting exhibit similar trends of improvement over unweighted embeddings, indicating that the granularity of weighting does not significantly affect predictive capacity.

In response to RQ3, TF/TF-IDF term weighting in static embeddings notably enhances performance compared to unweighted baselines, with weighted static embeddings often rivalling the performance of more resource-intensive contextual embeddings in limited data scenarios. This finding suggests that incorporating distributional information into static embeddings can be a cost-effective alternative to contextual embeddings, offering comparable performance while being computationally more efficient. Weighted static embeddings are thus particularly suitable for scenarios where computational resources are limited,

Figure 7.1: Spearman's ρ correlation, for the Amazon dataset, between the normalised frequency of the closest term distances (b_0) with a variable number of bins. ρ is reported in absolute magnitudes, higher is better. Dashed lines represent the baseline average of cosine distances across term-to-term comparisons, colours correspond to their respective experimental settings.



or when the trade-off between performance and efficiency favours the latter. The similar performance of domain- and document-level weighting further implies that the choice between these strategies is not critically impactful. Overall, the findings strongly advocate for integrating term weighting into static embeddings for transfer learning, balancing cost-efficiency with competitive performance.

7.6 Distributive Contextual Embeddings Evaluation

This analysis explores Distributive Contextual Embeddings (DCE), which aggregate contextual representations of terms within target-specific vocabularies. We compare DCEs against both contextual BERT embeddings and static, probability-weighted embeddings (using the TF-IDF, **D** weighting scheme) in terms of their ability to predict relative transfer gain. DCEs are unique because they involve direct comparisons between individual terms and compile these comparisons to form a broader domain representation. We expect that DCEs might offer stronger correlations over both unweighted and weighted, domain-level aggregated document vectors. This expectation is based on the premise that DCEs, by concentrating on individual term relationships, could capture finer details of how language is used within a specific domain.

In our analysis with Distributive Contextual Embeddings (DCEs), we use a histogram/bin

approach to sort the distances between terms into different categories. By dividing these term distances into bins, we can see which terms are closer or further apart in different domains. The number of bins we choose affects how detailed our analysis is. More bins mean we look more closely at small differences in term distances, while fewer bins give us a broader view.

Figure 7.1 shows this analysis for the Amazon dataset. We use Spearman's ρ to measure how the frequency of the closest term distances (b_0) relates to transfer learning performance, using different numbers of bins. A higher ρ value means a stronger relationship. The dashed lines are the average of cosine distances for term comparisons, used as a standard to compare against. The colours represent different experimental settings. From this analysis, we find:

- **In zero-shot settings, increasing the number of bins increases the strength of correlation**. Increasing the number of bins up to 20 strengthens the correlation. This means that looking closely at the smallest differences in term distances helps predict transfer learning success in this setting.
- **In few-shot and limited settings, correlation strength increases then decreases after a certain threshold**. The best results come from using 6 bins in few-shot and 8 bins in limited settings. After these points, more bins don't help and can even reduce the strength of the correlation.

These findings make it clear that choosing the right number of bins is key to the effectiveness of our analysis. The right number helps us accurately identify patterns in term usage across different domains. Specifically, in zero-shot settings, a higher number of bins (up to 20) gives us a clearer picture and better predicts transfer learning success. However, in few-shot and limited settings, a moderate number of bins (6 and 8, respectively) is more effective. This indicates that while detailed analysis is useful in some scenarios, too much granularity can be less helpful in others.

Table 7.4 presents an in-depth analysis using Distributive Contextual Embeddings (DCE). It focuses on how various entropy and diversity measures, applied to cosine distance distributions between term vectors, correlate with relative transfer gain. This table includes normalised frequencies, Shannon and Rényi entropies, Simpson's Index, and statistical moments like Mean, Variance, Skewness, and Kurtosis. Our aim is to explore how these different measures reflect the effectiveness of transfer learning across datasets and settings. We use Cosine Distance ($DOM - DOM$) as the divergence measure between baseline representations (BERT (C), BERT (S)†), and the average of cosine distances between term vectors, Avg. Cos. The measures are tested in the context of different datasets, abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). We examine the first (b_0) and last bin (b_{-1}) of the histogram distribution, representing the normalised

Table 7.4: Spearman's ρ correlations between DCE representation divergence and intermediate-to-target F_1 scores are presented. Datasets are abbreviated as AM (Amazon), YL (Yelp), YH (Yahoo), and IS (TREC-IS). BERT (C), BERT (S), and Avg. Cos. represent contextual/static BERT embeddings and average cosine distances as baselines. Histograms have fixed bin widths of 20, 6, and 8 for Zero-shot, Few-shot, and Limited settings, respectively. b_0 and b_{-1} refer to the normalised frequencies of distances in the first and last bins of the histogram distribution, respectively. $H(P)$ and $H_\alpha(P)$ represent Shannon and Rényi entropies; D represents Simpson's Index; μ, σ_2, $\tilde{\mu}_3$, and $\tilde{\mu}_4$ represent Mean, Variance, Skewness, and Kurtosis. Statistically significant correlations are denoted by an asterisk (*) after Bonferroni correction for the number of measures tested ($p < 0.005$). The **bold** values highlight the strongest correlations for each dataset and setting within the Histogram Analysis, best overall scores for each dataset/setting are underlined. Symbols ▲ and ▼ compare values to baseline (grey rows) within each measure, representing higher or lower correlation magnitudes, respectively.

	Zero-shot				Few-shot				Limited		
Measure	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
BERT (C)	**-0.904***	-0.787*	-0.787*	**-0.766***	0.272*	-0.063	0.435*	0.007	0.489*	0.105	0.497*
BERT (S)†	-0.847*	**-0.842***	**-0.809***	-0.554*	0.307*	-0.010	0.405*	0.077	0.506*	0.164	0.447*
Avg. Cos.	-0.415*	-0.421*	-0.568*	-0.401*	0.416*	0.517*	0.797*	0.062	0.589*	0.468*	0.779*
Histogram Analysis											
b_0	0.682*	0.578*	0.727*	**0.651***	-0.429*	**-0.535***	-0.766*	-0.051	-0.597*	-0.588*	-0.727*
b_{-1}	-0.329*	-0.339*	-0.498*	-0.365*	0.412*	0.525*	**0.826***	0.065	0.571*	0.442*	0.798*
$H(P)$	**-0.769***	-0.579*	-0.722*	-0.352*	0.403*	0.534*	0.668*	0.044	0.538*	0.579*	0.643*
$H_{\alpha(P)}$	-0.768*	-0.579*	-0.723*	-0.350*	0.402*	**0.535***	0.668*	0.044	0.537*	0.579*	0.640*
D	**-0.769***	-0.582*	**-0.726***	-0.306*	**0.431***	**0.535***	0.752*	-0.076	**0.590***	0.589*	0.716*
μ	**0.769***	0.582*	**0.726***	0.305*	**-0.431***	**-0.535***	-0.752*	0.076	**-0.590***	-0.590*	-0.716*
σ^2	0.639*	0.235*	0.734*	0.089	-0.408*	0.514*	-0.642*	0.089	-0.581*	0.436*	**-0.800***
$\tilde{\mu}_3$	-0.163*	-0.581*	-0.717*	-0.137*	-0.349*	**0.535***	0.763*	-0.066	-0.356*	0.590*	0.721*
$\tilde{\mu}_4$	-0.248*	**0.583***	0.377*	0.049	0.394*	**-0.535***	-0.761*	0.074	0.514*	**-0.592***	-0.718*
Alignment Cost Analysis											
Wass-1	0.492*	0.510*	0.654*	0.447*	-0.415*	-0.502*	-0.712*	-0.056	-0.598*	-0.494*	-0.709*

† BERT (S) is weighted by the TF-IDF, **D** weighting scheme from the previous experiment.

frequencies of the closest and farthest term distances, respectively. This approach helps us understand the range of term-to-term distances within a domain and their implications for transfer learning success. Statistical significance is denoted with an asterisk (*) after correcting for the number of measures tested ($p < \frac{0.05}{10} = 0.005$). The strongest correlations for each dataset and setting within the histogram analysis are highlighted in bold, and the best overall scores are underlined. Additionally, symbols ▲ and ▼ indicate whether the results show higher or lower correlation magnitudes compared to the highest values among the baselines, represented by greyed rows. From Table 7.4, we observe:

- **Histogram-based measures outperform averaged cosine distances**. Histogram-based measures, notably the frequency of distances in the first bin (b_0), demonstrate superior performance over averaged instance-to-instance cosine distances across most datasets and settings. This suggests that focusing on the closest term distances within histograms provides a more informative representation of domain similarities than considering the average distance between all term pairs. By emphasising the most similar term relationships, histogram-based measures better capture the po-

tential for transferability estimation, resulting in stronger correlations with transfer gain.

- **There are strong correlations with entropy and statistical moments in zero-shot settings**: In zero-shot settings, entropy measures (Shannon and Rényi entropies) and Simpson's Index show strong negative correlations. While these measures may not always surpass previous baselines, they remain competitive. The results suggest that domains with more consistent term distances (lower entropy) correlate with better transfer learning outcomes. Similarly, the mean and variance of these distributions are strongly correlated, highlighting their importance in predicting transfer success.

- **DCEs excel in few-shot and limited Settings**: DCEs achieve notably strong correlations in challenging few-shot and limited settings, often greatly exceeding previous baselines. This indicates DCEs' capability to distinguish between domains where other methods struggle, particularly in data-scarce situations. In few-shot settings, histogram-based measures outperform the best BERT baselines by significant margins in Amazon, Yelp, and Yahoo datasets by 0.124, 0.535, and 0.421, respectively. Similarly, in limited settings, there is a marked increase in correlation strength of 0.084, 0.124, and 0.282 for these datasets. This shift in performance, especially in scenarios where previous methods underperformed, underscores the effectiveness of DCEs in predicting transfer gain in difficult settings.

In conclusion, the analysis shows that DCEs are highly effective in predicting transfer learning performance, especially in few-shot and limited data scenarios where other methods have struggled. They significantly outperform both static and contextual embeddings, with histogram-based measures providing clear insights into term relationships and domain similarities. This superior performance in previously challenging settings highlights the practical value of DCEs in transfer learning, offering a reliable method for assessing transfer gain.

7.7 Representation Configuration Summary

This section provides a summary of the most effective embedding representation configurations discussed in this chapter. The aggregation methods that correlated most strongly with relative transfer gain for each type of distribution, along with specific representation configurations, are detailed. These configurations will be used in the feature engineering stage for the intermediate task selection process discussed in Chapter 8.

BERT Embeddings. BERT embeddings achieved their best performance with the contextual variant (BERT (C)), using the $DOM - DOM$ aggregation method and Cosine distance as the divergence measure.

Static BERT Embeddings. Weighting static BERT embeddings with document-level probabilities derived from TF-IDF distributions significantly enhanced performance compared to their unweighted counterparts. Moreover, results showed that weighted, static embeddings can rival more computationally expensive contextual embeddings.

Distributive Contextual Embeddings. In Distributive Contextual Embeddings, setting the number of histogram bins to 20, 6, and 8 for Zero-shot, Few-shot, and Limited settings, respectively, correlated strongly with relative transfer gain. The most effective measures from our histogram analysis were the averaged term-term cosine distances (Avg. Cos.), the normalised frequencies of the lowest term-term distances from the first bins (b_0), the Shannon entropy ($H(P)$), Simpson's Index (D), and the Expected Value (μ) of term-term distances. Additionally, the Wasserstein-1 measure used in our Alignment Cost Analysis provided competitive performance, demonstrating its effectiveness in this context.

In the following section, we will conclude this chapter with a discussion of our findings on embedding representations and how these findings contribute to our thesis, offering a comprehensive summary and reflecting on the implications of these results.

7.8 Conclusions

In this chapter, we continued our exploration into embedding representations, aiming to improve transferability estimation in natural language processing. Our focus was on three types of embeddings: BERT-based embeddings, Probability-Weighted Embeddings (PWEs), and Distributive Contextual Embeddings (DCEs). Each type underwent detailed evaluation to assess its predictive power in various domains and experimental settings.

Our research, as elaborated in Section 7.3, revolved around four primary research questions, guiding us through four separate but interconnected experiments. We evaluated BERT embeddings, investigating the effectiveness of both their general-purpose (static) and contextual embeddings. For PWEs, the assessment centred on how domain-specific frequency information, derived from our earlier work with Term Distributions, could enhance term embeddings. For the DCEs, we introduced a novel approach to embedding representation. This involved comparing divergences in term usage across different contexts, where we sought sought to understand how these contextual differences could better capture relationships between related domains.

Through experimentation in Sections 7.4, 7.5, and 7.6, our findings revealed that there was a clear relationship between the divergence of domain embedding representations and transfer learning performance. In particular, we found, unsurprisingly, that contextual BERT embeddings generally showed stronger predictive capabilities for transfer learning than static embeddings. Although we hypothesised that the semantic and syntactic information encoded in high-dimensional embeddings would result in improved performance in instance- and centroid-based comparisons, we often found that simpler, domain-level comparisons often surpassed or matched these comparisons in effectiveness, with the added benefit of computational efficiency. We found that, in the assessment of weighted embeddings (PWEs), integrating term frequency data into embeddings, PWEs demonstrated enhanced performance. Indeed, weighting even general-purpose, static BERT embeddings often matched or surpassed contextual BERT embeddings, offering an effective and practical solution to embedding construction. DCEs, combining insights from both distributional and embedding-based approaches, showed promising results in few-shot and limited data scenarios. Indeed, DCEs were the strongest predictor of domain divergence in difficult scenarios. Their ability to capture contextual differences in term usage in these difficult settings—often reporting strong to very strong correlations in settings which other representations reported no correlation—emphasises their potential for use in transferability estimation.

The results from this chapter further contribute to answering our first thesis research question (RQ1 in Section 1.3) by exploring the effectiveness of embedding-based representations in capturing domain characteristics and estimating transferability. Our experiments show that contextual BERT Embeddings, Probability-Weighted Embeddings (PWEs), and Distributive Contextual Embeddings (DCEs) demonstrate strong predictive capabilities for transfer learning. These findings reinforce our hypothesis that advanced domain representations and precise divergence estimations can significantly improve the accuracy of transferability estimation and optimal task selection in transfer learning.

In aligning with our thesis statement (see Section 1.2), these findings reinforce the notion that a systematic and well-informed approach to task selection, underpinned by sophisticated domain representations and divergence calculations, can enhance the accuracy of performance prediction in transfer learning. In the following chapter, we assess the practical application of the representations investigated thus far. Specifically, we will use the best configurations and divergence measures identified from both chapters to evaluate their effectiveness in intermediate task selection, examining the performance and efficiency of our approaches through numerous experiments.

Chapter 8

Effective Intermediate Task Selection

8.1 Introduction

This chapter focuses on the application of the distributional and embedding representations from previous chapters for selecting intermediate tasks in transfer learning. The primary objective of this chapter is to use the divergence estimations between these representations to accurately predict the ranking of tasks based on relative transfer gain. Our methodology uses divergence measure outputs as input features for regressors and rankers. For each target domain, we train a model on all other intermediate and target domain pairs (excluding instances where the intermediate domain, $\mathcal{D}_I$, is the same as the target domain, $\mathcal{D}_T$ to prevent bias). The evaluation of these models is conducted on instances where $\mathcal{D}_T$ is the target domain, resulting in a ranked list of intermediate tasks for each target.

In the practical use of our framework, the value add to an end user is being recommended the best intermediate-target task pairing to use for transfer. As such, the effectiveness of our approach is measured by the closeness of our estimated rankings to the actual performance rankings, with a secondary focus on the accuracy of true performance estimation (i.e. in terms of intermediate-to-target F_1 performance). Moreover, we consider that the user has a particular training budget, K, where each task pair that is tried incurs further costs in model training. Given the inherent uncertainty in model estimations, our methodology adopts a top-K ranking approach, where users can explore multiple recommended models, starting with the most promising ones. Ideally, the best task to use for transfer is recommended first, i.e. at the top rank, but providing a range of options up to a reasonable rank K ensures that users have viable alternatives. We report the ranking quality (NDCG [43] and Regret [86], discussed further in Section 8.2.3) across all values of K as a measure of the overall performance of our system; however, we focus primarily on the ranking quality at ranks 1-5. This decision is also informed by dataset constraints, where the maximum K for the dataset with the smallest number of domains

(Yahoo) is 9. As such, suggesting that the user explore more than half of the search space would not only be impractical but also defeat the purpose of efficient task selection. By concentrating on the top 1-5 ranks, we ensure that our evaluation aligns with realistic usage scenarios, where users seek to identify the most relevant tasks quickly.

In addition to evaluating the effectiveness of our system, the efficiency of our framework is equally important. As such, we assess the efficiency of producing each representation in terms of computational power, measured in kilowatt-hours (kWh), and end-to-end runtime, which includes both the time taken to construct and evaluate representations and the runtime of the models themselves. Furthermore, recognising the growing importance of environmental sustainability in computational practices, we also consider the associated environmental costs saved using our framework, quantified in terms of CO_2 emissions.

The analysis of our framework is structured to evaluate each category of representation—such as TD, LFD, PWE, and DCE—integrated as a single feature set, grouped by their respective chapters (Distributional or Embedding), and at an individual level. This involves comparing and contrasting various representations to determine which offers the most significant performance gains relative to resource consumption. In our comparative assessment, we seek to answer the following questions that directly answer our broader research objectives: Which representations provide the best balance between recommendation accuracy and resource utilisation? Are more computationally intensive representations justified in terms of their performance benefits? How do these choices align with the broader goal of sustainable development in transfer learning? The answers to these questions will offer valuable insights into the practical applicability of our framework in real-world scenarios, where both performance and efficiency are core concerns. The remainder of this chapter is structured as follows:

1. In Section 8.2.1, we outline the configurations—such as vocabulary choice for term distributions, aggregation method, among others—for each representation used for task selection. We also discuss any descriptive, diversity- and statistical moments-based features used.

2. Section 8.2.3 discusses our evaluation metrics used for evaluating both performance and efficiency.

3. In Section 8.2.2, we outline the models used for regression and ranking tasks, alongside details on setting hyperparameters, and additional feature or label transformations performed.

4. Sections 8.2.4, 8.2.5, and 8.2.6 outline our experimental methodology for performance, efficiency, and performance-efficiency experiments.

8.2 Methodology

This chapter's methodology is structured to systematically evaluate the effectiveness and efficiency of distributional and embedding representations in selecting intermediate tasks for transfer learning. The methodology is divided into six distinct subsections, each of which addresses the core aspects of the approach.

8.2.1 Representation-specific Configurations

This subsection details the specific representations used in our experiments, including their settings and configurations. The representations encompass a range of distributional and embedding-based types, as developed and refined in Chapters 6 and 7. For each representation type, we have selected the most fitting divergence measures and levels of aggregation, based on the insights gained from their respective analyses.

Baselines. Following prior work [72], our system's performance is benchmarked against two baseline approaches. The first is a random ranking of intermediate tasks, denoted as *Random*, which we average across 50 different seeds to reduce variance in the results. The second baseline involves using domain-level aggregated Sentence-BERT [85] embeddings (*SEmb*) with linear regression.

Distributional Representations. We use all distributional representations discussed in Chapter 6: Term Distributions (TD), Linguistic Feature Distributions (LFD), and Topic Frequency Distributions (KFD). The divergence measures and the aggregation level at which divergence is computed are as follows:

- **Term Distributions**: We use the Cosine distances and Rényi divergences computed between the TF and TF-IDF distributions using target-focused ($\mathcal{D}_T$) vocabularies. To maximise efficiency, we use TF- and TF-IDF-based term ranking methods for TF and TF-IDF distributions, respectively.

- **Linguistic Feature Distributions**: Across all LFDs, we use Bhattacharyya distances between the domain-aggregated ($DOM - DOM$) distributions. For distributions built from named entities (NER) and linguistic dependencies (DEP), we use divergence computed using Maximum Mean Discrepancy (with the Gaussian kernel) between randomly sampled instances ($INST - INST$), as our analysis (see Section 6.5.1) found that this yielded similarly strong correlations. Likewise, for distributions based on parts-of-speech tags, U/XPOS, we compute divergence at the domain level, using Maximum Mean Discrepancy with the Laplacian kernel and Central Moment Discrepancy.

- **Topic Frequency Distributions**: We use the divergence measure that, from our analysis in Section 6.6, resulted in the strongest correlation with intermediate-to-target F_1 performance scores, Wasserstein-1.

Embedding Representations. The embedding representations used in this chapter are BERT embeddings (BERT (C)), Probability-Weighted Embeddings (PWE), and Distributive Contextual Embeddings (DCE), using the following configurations:

- **BERT Embeddings**: In Section 7.4, we found that the contextual variant of our BERT embeddings resulted in the highest correlation with performance. Among these results, computing divergence using cosine distance at the domain level ($DOM-DOM$) resulted in the strongest correlation with our relative transfer gain.
- **Probability-Weighted Embeddings**: From our analysis, we found that weighting static embeddings by their TF-IDF-based, document-level probabilities resulted in strong correlations with performance, sometimes outperforming their unweighted, contextual counterparts. Therefore, we use the divergence between these PWE representations computed by the cosine distance.
- **Distributive Contextual Embeddings**: For DCE, we use measures from our analysis that correlate strongly with performance. We use the results from our baseline, which is the average of the cosine distances between the term vectors. From our histogram-based results, we set the optimal number of bins (as shown in Figure 7.1) to 20, 6, and 8 for zero-shot, few-shot, and limited settings and use: the normalised frequency count of the first bin, i.e. the closest term distances (b_0); the (Shannon, $H(P)$) entropy between the cosine distances; the Simpson's Index (D) between the cosine distances, and the expected value (μ) between them.

Alongside measures that describe the divergence between representations, we also incorporate metrics that quantify the intrinsic diversity within individual distributions, adhering to methodologies established in previous research [91]. This includes the evaluation of both Shannon and Rényi entropies, and distributional characteristics such as mean, variance, skewness, and kurtosis. Specifically, we apply these diversity metrics to characterise each intermediate representation across all intermediate and target pairs.

8.2.2 Models for Task Selection

In our approach to task selection in transfer learning, we address it as both a regression and ranking problem as our aim is to predict performance scores and effectively determine the order of tasks. The following models were chosen for the task selection experiments:

- **Regression**: In our analysis, we use the divergence between datasets as input features. Theoretically, we posit an inverse linear relationship, i.e. that an increase in model performance corresponds to a decrease in divergence between datasets. This makes Linear Regression a suitable choice as a computationally efficient and simple model for regression tasks. We complement the use of Linear Regression with the gradient-boosted ensemble model, XGBRegressor [17]. XGBRegressor is particularly adept at dealing with non-linear feature interactions, such as the interactions between model performance and our intermediate task diversity features.

- **Ranking**: Task selection in transfer learning has an objective similar to that of a recommendation system, where the objective is to "recommend" the most suitable intermediate task for transfer. Ranking models such as LambdaRank [14] and LambdaMART [15] are adept at sorting items (in this case, tasks) in order of relevance or suitability, making them ideal for this application. These models excel in identifying and prioritising the tasks that are most likely to yield significant performance benefits. LambdaRank, a pairwise ranker, is particularly effective at differentiating between pairs of tasks based on their relative ordering. We complement LambdaRank with LambdaMART, expecting that its ensemble of decision trees will enhance the model's ability to capture complex and non-linear relationships between features. This makes LambdaMART particularly adept at handling scenarios in which the effectiveness of a task for transfer learning is influenced by several complex feature interactions.

Each model was chosen to address the specific facets of task selection in transfer learning. They are designed to complement each other, providing a robust framework for predicting performance scores and optimising task order based on their predicted effectiveness. To fine-tune these models for optimal performance, a Bayesian search over the hyperparameter space is conducted, detailed in Appendix A.2. This search was performed using a set of five randomly selected tasks from the datasets considered.

8.2.3 Evaluation Metrics

Evaluation of intermediate task selection requires a method that accurately reflects the transfer gain/loss of different task combinations. To achieve this, we use two metrics: Normalised Discounted Cumulative Gain (NDCG) and Regret metrics (defined in Sections 2.3 and 2.6).

For regression tasks, where we aim to directly predict the F_1-score of each intermediate-to-target ($\mathcal{D}_I - \mathcal{D}_T$) model, we evaluate the outputs using NDCG to determine if the ordering of our predictions is close to the ordering of ground truth performance scores. To ensure a balanced and realistic comparison, we normalise our ground truth performance

scores to a range between 0 and 1, where 1 represents the best model performance (see Section 3.7 for more details).

To evaluate ranking models, we implement a graded scoring system based on the relative transfer gain of the models. This scoring system ranges from 0 to 4, depending on the model's effectiveness in the transfer learning context.

- **Maximum Transfer Gain (Score: 4)**: Assigned to the model with the highest F1-score, this score reflects the maximum transfer gain (i.e. attainable F_1-score performance) within the target domain list.
- **High Transfer Gain (Score: 3)**: Models ranked second and third are given a score of 3, denoting significant but slightly lesser transfer gains compared to the top-ranked model.
- **Moderate Transfer Gain (Score: 2)**: The fourth- and fifth-ranked models receive a score of 2, recognising their value in transfer learning while differentiating them from the top performers.
- **Low Transfer Gain (Score: 1)**: Models ranked beyond the fifth position down to the bottom 30% of the list are assigned a score of 1. This category acknowledges their role in transfer learning, albeit with lower gains.
- **Minimal Transfer Gain (Score: 0)**: Models in the bottom 30% are given a score of 0, indicating minimal or negligible transfer gain, thus signalling their limited utility in transfer learning.

This grading provides a quantitative framework for identifying the relative transfer gain between models, that is, for identifying tasks that are most effective in leveraging knowledge from the intermediate task. This grading is also important in guiding the ranking algorithms. By assigning distinct scores to different tiers of model performance, we instruct the algorithm on how to prioritise models based on their effectiveness.

8.2.4 Performance Evaluation

The evaluation of task selection performance in this chapter involves training models on combinations of intermediate and target domains, $\mathcal{D}_I$ and $\mathcal{D}_T$, respectively. We exclude cases where $\mathcal{D}_I = \mathcal{D}_T$ to avoid overfitting bias, ensuring that our models do not simply learn domain-specific patterns that do not generalise to other domains. This approach yields a ranked list of intermediate domains for each target domain.

Our initial experiments involve comparing each of our models—Linear Regression, XGBRegressor, LambdaRank, and LambdaMART—against a random ranking and a Sentence

Embeddings (SEmb) baseline, following prior work [72] on intermediate task selection with adapters. This comparison establishes a baseline for our model's performance. We use two main metrics for evaluation: NDCG and Regret. NDCG measures the quality of the rankings produced by the models, whereas Regret quantifies the performance gap between the model and an optimal scenario. We report Regret at rank cut-offs of 1 and 3, following the methodology used by Poth et al. [72] and focusing on ranks associated with the highest transfer gains according to our grading system. We also use the performance scores from this table to inform the model choice for subsequent experiments. Specifically, for each dataset-experimental setting combination, we select the model with the highest NDCG performance for that setting.

Following the baseline experiments, we conduct a more detailed domain-specific evaluation. We analyse how maximal performance (measured using the F_1-score) is distributed across different rank intervals within a set of domains. This involves examining the number of domains achieving their highest F_1-score within the top-1, top-3, and top-5 rankings, including those beyond this threshold. This analysis, complemented by baseline metrics, offers a comprehensive view of the performance of our models. The visualisation of these results is presented in a bar chart displaying the number of domains reaching their maximal F_1-score within the top-1, top-3, and top-5 ranks, as well as those outside these ranges, for each dataset and experimental setting. This approach allows us to see where each model excels or falls short, offering clear, domain-specific performance insights that the broader baseline metrics do not express.

We then explore the impact of different feature categories on model performance by training models using the following feature sets derived from each representation category: Term Distributions, Linguistic Feature Distributions, Topic Frequency Distributions, Contextual BERT Embeddings, Probability-Weighted Embeddings, and Distributive Contextual Embeddings. The aim of this experiment is to isolate and understand the contribution of each representation type to the overall model efficacy. For distributional measures, we complement the features based on divergence with features that describe the diversity (entropy-based and statistical moments-based measures) of intermediate task distributions. We evaluate these models using NDCG@K, offering a direct comparison of the effectiveness of each feature category and highlighting their strengths and weaknesses in different experimental settings, relative to each other and to the Sentence Embeddings (SEmb) baseline. This comparison not only serves as a benchmark and helps assess the relative improvement offered by each feature type over a cost-effective alternative.

8.2.5 Efficiency Evaluation

In the Efficiency Analysis section of this chapter, we present a comprehensive cost-based breakdown of our framework. This includes reporting efficiency metrics related to the

construction and analysis of representations. We used the CodeCarbon [55] package to record the efficiency metrics. Specifically, we focused on three key aspects:

- **Runtime (RT)**: This metric indicates the wall-clock time, measured in minutes, required to construct the representations for each target domain.
- **CO_2 Equivalents (CO_2eq)**: Here, we report the carbon footprint, in grams of CO_2 equivalents, associated with constructing these representations. This metric is important for understanding the environmental impact of our computational processes.
- **Energy Consumption (kWh)**: This metric shows the total energy consumption in kilowatt-hours for the construction and analysis of the representations.

Our methodology involves recording the time needed, emissions emitted, and energy expended to construct these representations and compute the divergence between them. The values reported in the table represent the average cost of computing these representations for a single target domain. This means the cumulative effort required to construct and analyse representations for all intermediate tasks associated with that target domain, as well as the representation for the target domain itself. Additionally, for representations that require aggregation to approximate effective divergence, we include these aggregation costs in our calculations. We benchmark these costs against two points of reference:

- **Sentence Embeddings (SEmb)**: This comparison provides a cost analysis against a more cost-effective representation baseline.
- **Model Training**: We compare our representation construction costs with those of a more resource-intensive approach, where every intermediate model is trained and tested.

8.2.6 Evaluating Performance-Efficiency Trade-offs

We evaluate the balance between the performance of the models and the computational resources and time required to achieve this performance. Our focus is on practical application: aiding users in identifying the most effective model within a given budget or time constraint, represented by K. We consider a scenario in which users with a specific training budget K navigate through a ranked list of intermediate task candidates. As the user traverses the list, they increase the likelihood of identifying the most effective model by exploring more task combinations. To quantify this, we calculate the cumulative maximum F_1-score at each rank of the predicted ranking. This measure reflects the best performance achieved up to that point in the search. We also track the cumulative

runtime for each model up to a given rank. This includes the time taken to train intermediate models, additional fine-tuning on target training sets (in the few-shot and limited settings), and the time required for inference on the target test set.

The primary aim of this experiment is to demonstrate the time-saving potential of our framework. We compare our approach against two alternatives: the traditional full grid-search method, which involves training and testing all possible model combinations, and the divergence between Sentence Embeddings (SEmb) as a baseline estimator of intermediate task ranking. Our hypothesis is that our framework can significantly reduce the time and computational resources required to identify the optimal model.

We report the average values of F_1@K and Runtime@K to provide a realistic overview of the framework's performance in practical settings, where a user aims to optimise performance for a specific task.

8.3 Research Questions

RQ1: How effective are the distributional and embedding representations developed in our approach in surpassing baseline methods (random ranking and sentence embeddings) in terms of both ranking and transferability metrics across various datasets and experimental settings?

This research question evaluates the performance of our representations—both distributional and embedding—in the context of intermediate task ranking for transfer learning. The effectiveness of these representations is measured against two baseline methods: a random ranking approach and sentence embeddings. The evaluation considers two key metrics: the ability to correctly rank intermediate tasks (NDCG) and the subsequent transferability to target tasks (Regret). The comparison spans each dataset and experimental setting to ensure a comprehensive assessment of the effectiveness of our representations under diverse conditions.

RQ2: How effective is our task selection approach in achieving maximal F_1-score performance within the top 1, 3, and 5 ranks under different experimental settings (zero-shot, few-shot, limited)?

This question investigates the precision of our task selection method in identifying the most beneficial intermediate tasks for transfer learning. Specifically, it investigates how often the optimal task (yielding the highest F_1-score) is found within the top-1, top-3, or top-5 ranked tasks. By focusing on the top rankings, this question directly tests the core competency of our task selection approach: its ability to discern and prioritise the tasks that are most likely to enhance performance in transfer learning scenarios.

RQ3: How does the effectiveness of different types of representations compare in the context of NDCG@K performance, averaged across domains within each dataset, for ranking intermediate tasks?

This research question focuses on comparing different text representations in terms of their efficacy in ranking intermediate tasks. The metric used for this comparison is NDCG at various cutoffs (K), which evaluates the quality of the rankings produced. By averaging these NDCG@K scores across all domains within each dataset, we determine which representation method (distributional or embedding-based) consistently yields better rankings. This comprehensive comparison across domains and datasets allows us to determine the most effective representation type for intermediate task selection.

RQ4: What are the costs associated with constructing and analysing different representations for predicting intermediate tasks in terms of runtime, CO_2 emissions, and energy consumption?

This question seeks to quantify the resource costs involved in developing and using our representation methods for task selection. This involves calculating and comparing the runtime (in minutes), carbon dioxide (CO_2) emissions (in grams), and energy consumption (in kilowatt-hours) for each representation method. These metrics provide insight into the environmental and operational efficiency of our methods, which we compare with both the Sentence-BERT baseline and the cost of training and inference of all intermediate models.

RQ5: How does our task selection approach demonstrate efficiency in achieving maximal F_1-score performance compared with both a full grid search and baseline methods?

This research question investigates the dynamics between performance and time efficiency in our task selection approach. Specifically, it examines the progression of F_1-scores as we move down the list of ranked tasks (K) and how this correlates with the cumulative runtime. The key measure here is the point at which the maximum F_1-score is achieved for all domains relative to the cumulative runtime. This comparison with a full grid search and baseline methods underscores the effectiveness of our approach in balancing high performance with time efficiency.

RQ6: How effectively does our framework identify the maximal performance of each domain early in the task selection process, thereby saving time compared with a full grid search?

RQ6 focuses on the time savings achieved by our framework for each individual domain. This question highlights the extent of time saved in each domain by our approach compared

with a full grid search. The visualisation of time savings per domain emphasises the practical impact of our method in diverse settings, showcasing its ability to efficiently reach optimal performance and thus providing a detailed complement to the dataset-level analysis in RQ5.

8.4 Performance Analysis

This set of baseline experiments compares the performance of various models—Linear Regression, XGBRegressor, LambdaRank, and LambdaMART—against a random ranking and a Sentence Embeddings (SEmb) baseline. The objective of this experiment is to establish a performance benchmark for task selection models in three different settings: zero-shot, few-shot, and limited. We use NDCG and Regret as our primary evaluation metrics, with higher NDCG and lower Regret indicating better performance. These results will guide the selection of the most effective model for each dataset-experimental setting combination in subsequent experiments.

Table 8.1: Evaluation of intermediate task rankings produced by different methods are shown. The models used in our approach—Linear Regression (LR), XGBRegressor (XGBReg), LambdaRank, and LambdaMART—are compared against two baseline approaches—a random ranking of tasks (Random) and sentence embeddings (SEmb)—in mean *NDCG* and *Regret* scores across each dataset—Amazon (AM), Yelp (YL), Yahoo (YH), and TREC-IS (IS)—and experimental setting—Zero-shot, Few-shot, and Limited. The best score in each group (measure, dataset) is highlighted in bold. For *NDCG*, higher is better; for *Regret*, lower is better. Symbols ▲ and ▼ compare values to the best values among the baselines (grey rows), representing better or worse results, respectively.

		Zero-shot				Few-shot				Limited		
Method	Measure	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH
Random	NDCG	0.667	0.668	0.709	0.813	0.864	0.829	0.824	0.882	0.851	0.839	0.816
	R@1	0.754	0.683	0.606	0.334	0.093	0.081	0.085	0.122	0.024	0.017	0.014
	R@3	0.552	0.418	0.305	0.176	0.054	0.040	0.037	0.054	0.014	0.008	0.005
SEmb (LR)	NDCG	0.976	0.969	0.945	0.981	0.822	0.789	0.829	0.866	0.840	0.830	0.782
	R@1	0.054	**0.016**	**0.052**	0.024	0.138	0.083	0.082	0.161	0.026	0.016	0.016
	R@3	0.012	**0.007**	0.034	0.002	0.090	0.074	0.034	0.071	0.018	0.009	0.008
LR	NDCG	0.980▲	**0.972**▲	**0.962**▲	0.977▼	0.916▲	0.915▲	0.849▲	0.885▲	0.857▲	0.878▲	0.825▲
	R@1	0.055▼	0.054▼	0.087▼	0.052▼	0.050▲	0.032▲	0.048▲	0.119▲	0.022▲	0.015▲	0.013▲
	R@3	0.006▲	0.021▼	**0.000**▲	0.016▼	0.018▲	0.020▲	0.025▲	0.053▲	0.013▲	0.007▲	**0.003**▲
XGBReg	NDCG	**0.983**▲	0.953▼	**0.962**▲	**0.991**▲	0.928▲	0.899▲	0.806▼	0.913▲	0.868▲	0.848▲	0.821▲
	R@1	**0.049**▲	0.122▼	0.087▼	**0.012**▲	0.036▲	0.039▲	0.099▼	0.077▲	0.018▲	0.016–	**0.011**▲
	R@3	**0.000**▲	0.024▼	**0.000**▲	**0.000**▲	0.018▲	0.022▲	0.038▼	0.047▲	0.009▲	0.006▲	0.006▼
LambdaRank	NDCG	0.965▼	0.952▼	0.957▲	0.988▲	**0.964**▲	**0.932**▲	**0.875**▲	**0.929**▲	0.911▲	0.898▲	0.812▼
	R@1	0.083▼	0.093▼	0.069▼	0.022▲	**0.014**▲	0.026▲	0.066▲	**0.060**▲	0.012▲	0.009▲	0.014–
	R@3	0.014▼	0.023▼	0.008▲	0.001▲	**0.006**▲	0.010▲	**0.020**▲	0.028▲	**0.005**▲	0.003▲	0.007▼
LambdaMART	NDCG	0.961▼	0.946▼	0.947▲	0.983▲	0.956▲	0.924▲	0.872▲	0.920▲	**0.920**▲	**0.903**▲	**0.837**▲
	R@1	0.090▼	0.135▼	0.112▼	0.025▼	0.022▲	**0.021**▲	**0.045**▲	0.083▲	**0.010**▲	**0.004**▲	**0.011**▲
	R@3	0.015▼	0.024▼	**0.000**▲	0.001▲	0.006▲	**0.006**▲	**0.020**▲	**0.018**▲	**0.005**▲	**0.001**▲	0.006▼

Table 8.1 presents mean NDCG and Regret scores for each model across various datasets (Amazon, Yelp, Yahoo, TREC-IS) and experimental settings. The best scores for each measure are in bold. Symbols ▲ and ▼ indicate whether the model's performance

is better or worse than the best baseline. The table is interpreted by comparing the performance of our models against the baselines, focusing on NDCG for ranking quality and Regret for the gap from optimal performance. We observe the following:

- **Models generally outperform baselines**. Across various measures, our models consistently achieve higher scores than baselines, demonstrating their robust predictive capabilities in intermediate task selection. For NDCG, our models outperform the baseline in every dataset and setting. In terms of Regret metrics, our models surpass the baseline in 9 out of 11 cases for Regret@1 and 10 out of 11 cases for Regret@3.

- **SEmb performs strongly in zero-shot cases**: In zero-shot scenarios, the Sentence-BERT (SEmb) baseline excels in Regret-based metrics for Yelp and Yahoo datasets. Although SEmb closely matches our method in NDCG scores, our approach performs better in Regret metrics for Amazon and TREC-IS. This indicates SEmb's efficacy as a cost-effective option in zero-shot scenarios.

- **XGBRegressor and Linear Regression lead in zero-shot**: XGBRegressor consistently tops performance in zero-shot settings across most datasets and metrics, particularly in Regret@3. Linear Regression also shows strong results, indicating a linear relationship between divergence features and transfer gain in zero-shot cases.

- **Ranking models excel in few-shot and limited settings**: In more difficult settings, ranking models like LambdaRank and LambdaMART outshine simpler models. LambdaRank achieves the highest NDCG scores in all few-shot experiments, and LambdaMART tops NDCG in all limited experiments. LambdaMART also records the lowest Regret@3 scores, reinforcing the effectiveness of ranking models in challenging experimental settings.

- **Consistently high NDCG scores**: Our models maintain high NDCG scores across all settings, with scores above 0.9 in zero-shot and the lowest being 0.876 and 0.837 in few-shot and limited settings, respectively. This consistent performance highlights the reliability of our approach in producing quality rankings.

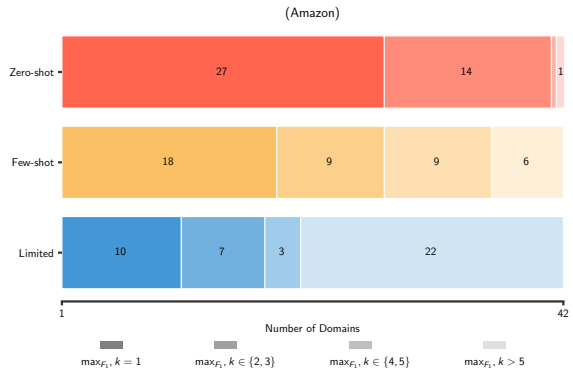
In response to RQ1, our models demonstrate superior performance over random ranking and SEmb baselines in most scenarios, confirming the strength of our representations in task selection. The consistently high NDCG scores across various settings emphasise our method's accuracy in ranking tasks. While SEmb remains a viable option in certain zero-shot cases, ranking models like LambdaRank and LambdaMART prove to be more effective in few-shot and limited scenarios. Conversely, simpler models like XGBRegressor and Linear Regression excel in zero-shot settings, suggesting a stronger linear relationship between

divergence features and transfer gain. This difference in model performance supports our rationale of approaching these experiments as both a regression and ranking problem. For the rest of this section, we choose the best-performing model for each dataset-experimental setting combination: XGBRegressor for Amazon and TREC-IS, Linear Regression for Yelp and Yahoo in zero-shot settings, and LambdaRank and LambdaMART for few-shot and limited settings, respectively.

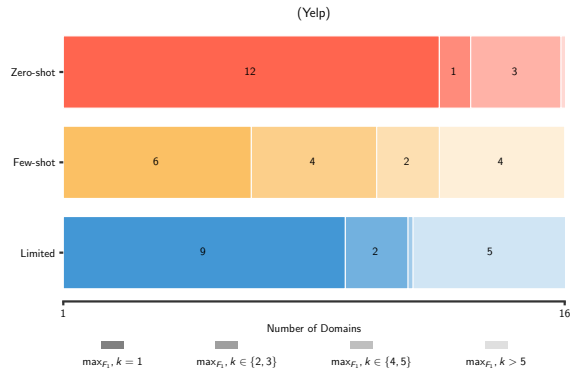
8.4.1 Rank Position Analysis for Maximum Domain Performance

This experiment evaluates our task selection method at the domain level by analysing the frequency with which the optimal model for a domain is ranked in the top positions. In line with our grading system in Section 8.2.3, we consider our approach to be successful if we are able to find the best-performing model in the top-5 ranks.

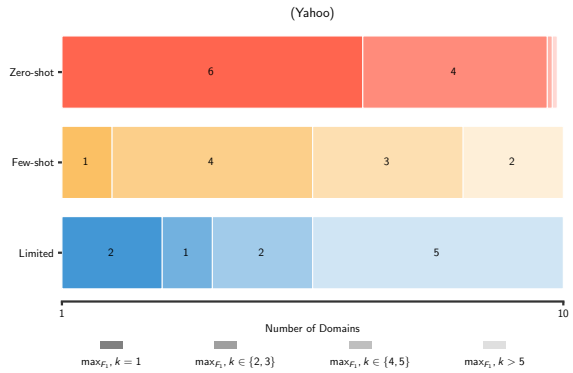
Figure 8.1: Bar chart displaying the number of domains achieving maximal F_1-score performance within top-1, 3, and 5 ranks (also incl. those that fall outwith this range), across various datasets. Each bar, colour-coded to signify different experimental settings (red for zero-shot, yellow for few-shot, and blue for limited), represents a distinct dataset. The opacity of each bar segment gradually decreases, indicating top-1, 3, and 5, where the most faint segment denotes $\max_{F_1}, k > 5$. Annotations on each segment provide a count of domains in both categories.



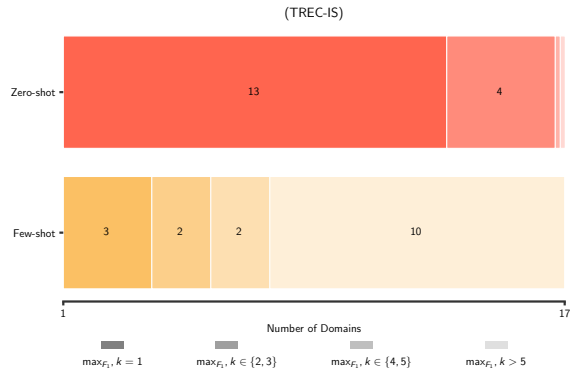
(a) $\max_{F_1}$ @k, Amazon.



(b) $\max_{F_1}$ @k, Yelp.



(c) $\max_{F_1}$ @k, Yahoo.



(d) $\max_{F_1}$ @k, TREC-IS.

Figure 8.1 visualises this analysis through a segmented bar chart for each dataset, with colours representing zero-shot (red), few-shot (yellow), and limited (blue) scenarios. The chart's density of each segment indicates the rank range, helping us determine how often the best model for a domain appears within the top ranks. Our key observations are:

- **It is easy to predict intermediate tasks for zero-shot settings**. In zero-shot scenarios, nearly all domains from all datasets rank within the top-5. Specifically, for Amazon, 27 out of 42 domains peak at rank 1, with another 14 in ranks 2 and 3. In Yelp, 75% reach maximum performance at rank 1, with the rest in ranks 2-5. For Yahoo and TREC-IS, all models are in the top-5, with 60% and 76.4% at rank 1, respectively. This underlines our method's efficiency in identifying the best models quickly in zero-shot scenarios.

- **Few-shot settings show a wider distribution of ranks**: In few-shot settings, despite a broader distribution of ranks, a significant portion of domains, especially in Amazon and Yelp, still achieve their best performance within the top-5 ranks. In Amazon and Yelp, 64.2% of domains peak in the top 3 ranks. Yahoo and TREC-IS show more spread, with 50% and 29.4% in the top 3, respectively. However, TREC-IS has 58.8% of domains reaching their best outside the top-5, indicating increased unpredictability with additional training data.

- **Limited settings are even more challenging**: Limited scenarios see a rise in domains where optimal performance occurs beyond the top-5 ranks, notably in Amazon and Yahoo. The proportions of domains outside of the top 5 ranks are 52.3% for Amazon, 56.3% for Yelp, and 20% for Yahoo, signifying the growing difficulty in accurately predicting the best models with additional fine-tuning.

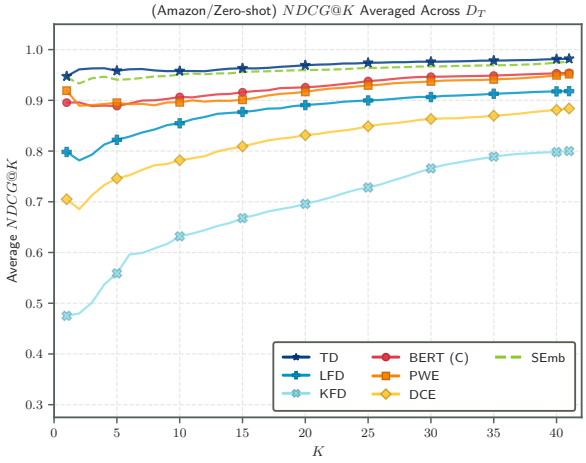
In answering RQ2, our methodology is highly effective in zero-shot settings, quickly finding optimal models for each domain. In few-shot settings, it remains effective, despite a broader spread in maximum performance. In limited scenarios, accuracy further decreases, highlighting the complexities of prediction with more domain-specific data. In the following set of experiments, we evaluate the performance of the features derived from individual representation categories from previous chapters.

8.4.2 Ranking Performance Analysis of Representations

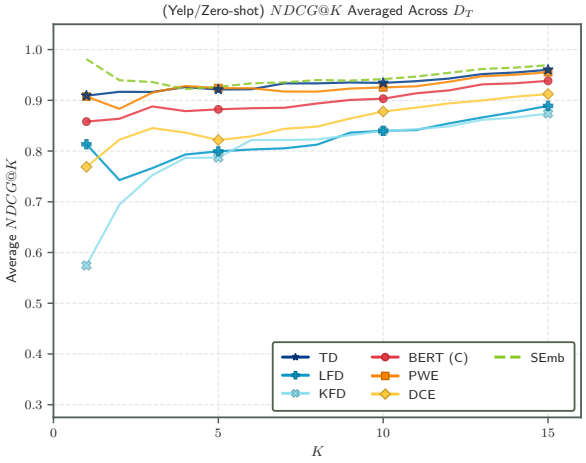
This experiment examines the impact of different representation categories on model performance by evaluating NDCG@K scores. The categories include Term Distributions (TD), Linguistic Feature Distributions (LFD), Topic Frequency Distributions (KFD), Contextual BERT Embeddings (BERT (C)), Probability-Weighted Embeddings (PWE),

and Distributive Contextual Embeddings (DCE). The goal of this experiment is to identify which representations demonstrate the greatest contribution to ranking intermediate tasks. Based on our conclusions from Sections 6.8 and 7.8, we anticipate that TDs, BERT (C), PWE will perform well in zero-shot settings while DCE will perform well in more difficult few-shot and limited settings. We focusing on the performance of representations in each dataset, dividing our analysis by experimental setting.

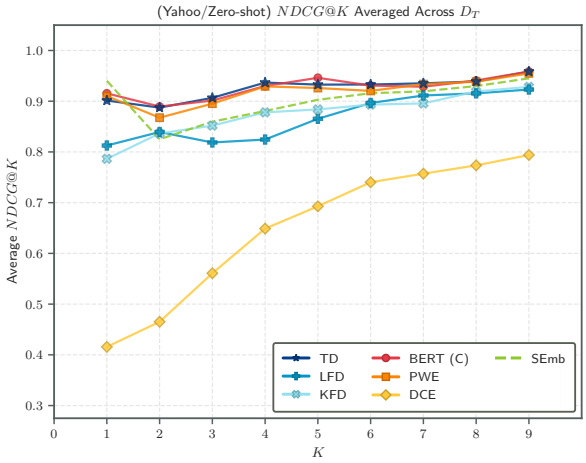
Figure 8.2: NDCG values averaged across each $\mathcal{D}_T$ for each dataset in the zero-shot setting, increasing in values of K. Higher values indicate better performance. The graph shows different distributional and embedding representation categories. Lines with cool blue tones denote distributional categories: Term Distributions (TD), Linguistic Feature Distributions (LFD), and Topic Frequency Distributions (KFD), complemented with markers—stars, plus signs, and crosses, respectively. Lines with warm tones (red, orange, yellow) denote embedding representations: Contextual BERT Embeddings (BERT (C)), Probability-Weighted Embeddings (PWE), and Distributive Contextual Embeddings (DCE), complemented with markers—circles, squares, and diamonds, respectively. The Sentence Embeddings baseline (SEmb) is represented by a green dashed line.



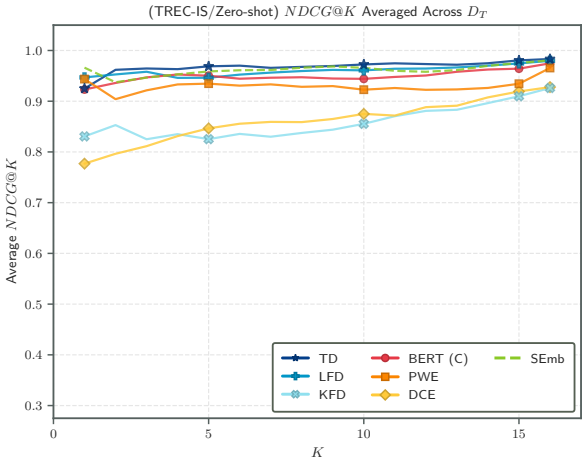
(a) NDCG@K for the Amazon dataset.



(b) NDCG@K for the Yelp dataset.



(c) NDCG@K for the Yahoo dataset.



(d) NDCG@K for the TREC-IS dataset.

Figure 8.2 shows the average NDCG@K values for different datasets in the zero-shot setting, with lines and markers indicating different representation categories. Distributional representations are represented by cool, blue hues. Embedding representations are represented by warm tones. Each of our representations are compared against the Sentence-BERT Embeddings (SEmb) baseline, denoted by a green, dashed line. As mentioned at the beginning of this section, we are primarily interested in the top-5 ranks, however, we report all values of NDCG for completeness. Observations from NDCG@K in zero-shot settings include:

- **Term Distributions (TD) generally have the highest NDCG@K values**. In zero-shot settings, TD outperforms other representations in the Amazon and Yelp datasets, especially at higher ranks, achieving an NDCG@1 value of 0.947 and 0.909 for both datasets, respectively. In Yahoo and TREC-IS datasets, they are often among the best representations, often close to the best-performing representations with a marginal difference.

- **Sentence-BERT (SEmb) is a competitive baseline**. Echoing findings from Chapter 7, SEmb closely matches TD in Amazon and even outperforms TD by 7.9% at NDCG@1 (0.981) for the Yelp dataset. Furthermore, SEmb outperforms PWE by 6.7% and 8% in Amazon and Yelp datasets, respectively, demonstrating its effectiveness as a representation.

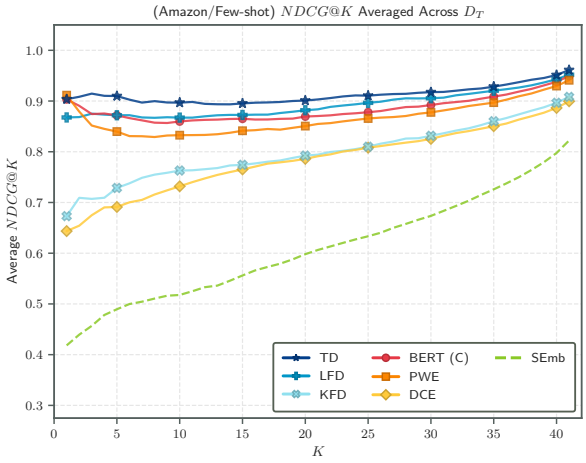
- **Probability-Weighted Embeddings (PWE) consistently outperform contextual BERT embeddings (BERT (C))**. PWE demonstrates comparable or better performance than BERT (C) in zero-shot settings. Since static embeddings are substantially cheaper to compute, they offer a cheaper and more effective alternative to resource-intensive contextual embeddings, supporting our findings from Section 7.8.

- **Topic Frequency Distributions (KFD) are weaker in zero-shot settings**. A recurring theme across all datasets is that KFDs tend to come last in zero-shot settings. Indeed, in the Amazon dataset, the Average NDCG@1 for KFDs is 49.8% less than TD. This suggests that, at least in zero-shot settings, a more granular comparison of frequencies (as opposed to a focus on broader themes in the text) is more beneficial.

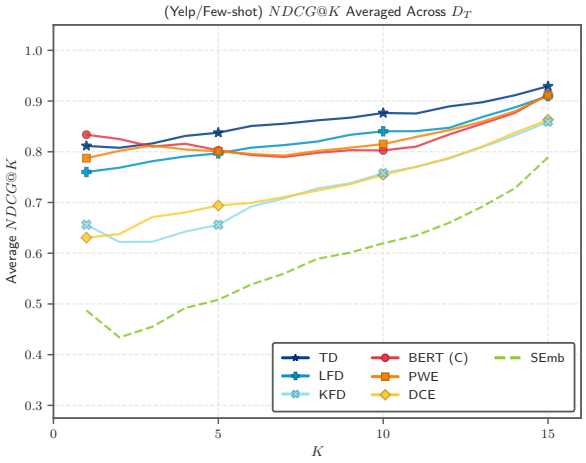
Figure 8.2 shows the average NDCG@K values for different datasets in the few-shot setting. The format of this figure is the same as in the zero-shot setting, with lines and markers indicating different representation categories. Key observations include:

- **The performance of SEmb drops in few-shot settings**: SEmb shows a significant performance drop in Amazon and Yelp compared to other representations, with

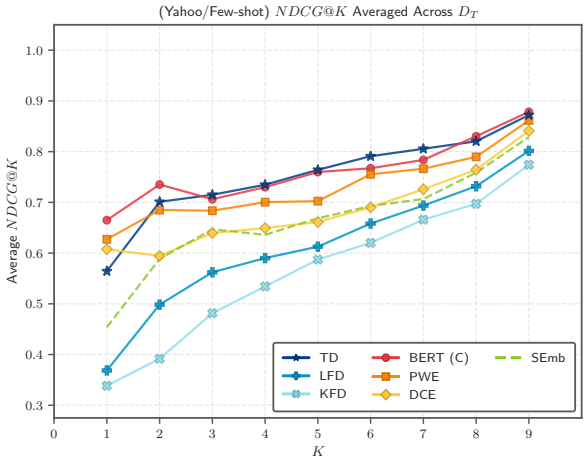
Figure 8.3: NDCG values averaged across each $\mathcal{D}_T$ for each dataset in the few-shot setting, increasing in values of K. Higher values indicate better performance. The graph shows different distributional and embedding representation categories. Lines with cool blue tones denote distributional categories: Term Distribution (TD), Linguistic Feature Distribution (LFD), and Topic Frequency Distribution (KFD), complemented with markers—stars, plus signs, and crosses, respectively. Lines with warm tones (red, orange, yellow) denote embedding categories: Contextual BERT Embeddings (BERT (C)), Probability-Weighted Embeddings using TFIDF-$\mathcal{D}$ weighting (PWE), Distributive Contextual Embeddings (DCE), complemented with markers—circles, squares, and diamonds, respectively. The Sentence Embeddings baseline (SEmb) is represented by a green dashed line.



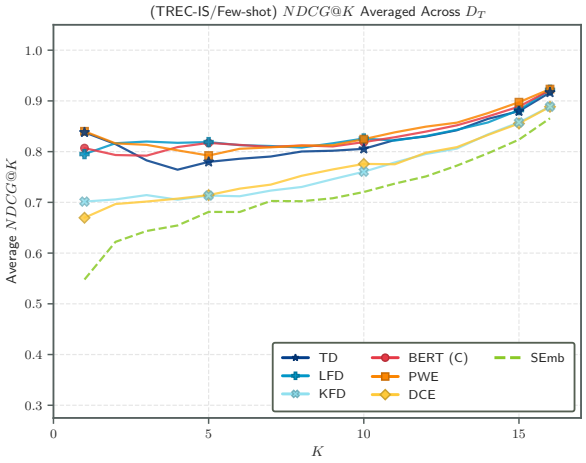
(a) NDCG@K for the Amazon dataset.



(b) NDCG@K for the Yelp dataset.



(c) NDCG@K for the Yahoo dataset.



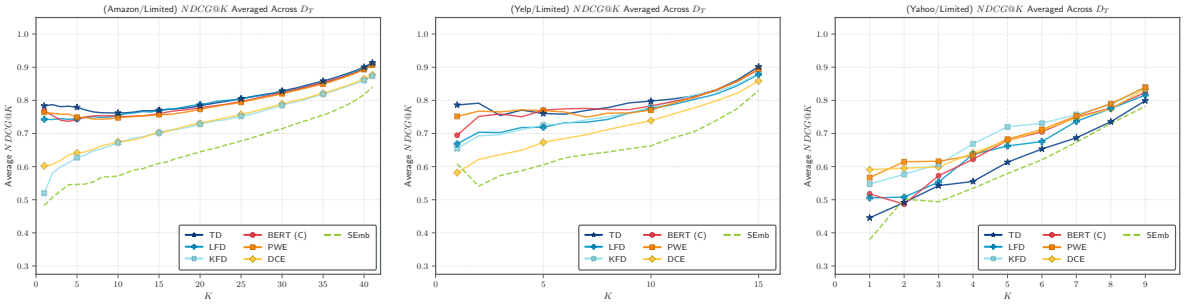
(d) NDCG@K for the TREC-IS dataset.

a 54.2% decrease compared to PWE in the Amazon dataset and a decrease of 41.6% when compared to the best-performing embedding. This performance shift suggests that while SEmb is effective in zero-shot scenarios, its adaptability to scenarios where there is further fine-tuning on limited data.

- **DCEs trail behind, but often outperform the baseline**. Out of our represen-

tations, DCEs typically come in last, with the exception of the Yahoo dataset, where DCE is third. This is somewhat unexpected due to the high correlation values in Section 7.6. This suggests that while DCEs correlate well with certain aspects of the data in these specific settings, this correlation does not directly translate into superior ranking performance.

Figure 8.4: NDCG values averaged across each $\mathcal{D}_T$ for each dataset in the limited setting, increasing in values of K. Higher values indicate better performance. The graph shows different distributional and embedding representation categories. Lines with cool blue tones denote distributional categories: Term Distribution (TD), Linguistic Feature Distribution (LFD), and Topic Frequency Distribution (KFD), complemented with markers—stars, plus signs, and crosses, respectively. Lines with warm tones (red, orange, yellow) denote embedding categories: Contextual BERT Embeddings (BERT (C)), Probability-Weighted Embeddings using TFIDF-$\mathcal{D}$ weighting (PWE), Distributive Contextual Embeddings (DCE), complemented with markers—circles, squares, and diamonds, respectively. The Sentence Embeddings baseline (SEmb) is represented by a green dashed line.



(a) NDCG@K for Amazon. (b) NDCG@K for Yelp. (c) NDCG@K for Yahoo.

Lastly, figure 8.4 shows the average NDCG@K values for different datasets in the few-shot setting. The format of this figure is the same as in the zero-shot and few-shot settings, with lines and markers indicating different representation categories. Key observations include:

- **SEmb further declines in performance**. SEmb shows a significant performance decrease in both Amazon and Yelp datasets. Compared to PWEs, SEmb reports a performance decrease of 36.9% and 33% for Amazon and Yelp datasets. This trend, continuing from few-shot into limited settings, underscores its challenge in adapting to these scenarios.

- **Performance in Yahoo is relatively weak**. The performance in the Yahoo dataset is substantially weaker compared to other settings, with values ranging from 0.380 to 0.547, implying a general difficulty for these representations to effectively adapt to the specific characteristics of this dataset.

In summary, our analysis across different settings reveals general trends in data representation performance. In zero-shot settings, Term Distributions (TD) excel, indicating their effectiveness without prior training. Sentence-BERT Embeddings (SEmb), while competitive in zero-shot, show a notable decline in performance in few-shot and limited settings. Probability-Weighted Embeddings (PWE) emerge as a cost-effective alternative, often surpassing more resource-intensive embeddings like BERT (C) in zero-shot scenarios. However, Topic Frequency Distributions (KFD) and Distributive Contextual Embeddings (DCE) consistently underperform, particularly in settings with further fine-tuning on target domain data.

8.5 Efficiency Analysis

This section presents an analysis of the efficiency of constructing and evaluating the representations used throughout this thesis. We frame our efficiency analysis around the average cost to construct representations to predict task selection for a single domain in each dataset, these include: (1) the costs of constructing the target domain representations themselves; (2) the cost of constructing intermediate domain representations—41, 15, 9, and 16 intermediate domain representations for Amazon, Yelp, Yahoo, and TREC-IS, respectively—to enable transferability estimation; (3) the associated costs of any advanced aggregation methods such as instance- and centroid-based calculations; and (4) representation-specific costs such as vocabulary construction for term distributions. Using the CodeCarbon [55] Python package, we track costs in terms of runtime, CO_2 emissions, and kilowatt-hours used to provide a comprehensive breakdown of the cost of our approach.

Table 8.2 provides a detailed cost breakdown for constructing representations for a single target domain in each dataset. RT denotes the total wall-clock time, recorded in minutes; CO_2eq denotes the CO_2-equivalents measured in grams; and kWh denotes the kilowatt-hours incurred as a result of producing and evaluating these representations. We also report the cost of constructing and evaluating Sentence-BERT (SEmb) embeddings to compare them to our representations, along with the average cost of training all of the intermediate models for a single domain. Key observations include:

- **LFDs and BERT (C) are the most resource-intensive**. For LFD, the wall-clock time ranges from 0.72 to 99.58 minutes, CO_2 emissions from 1.81g to 235.00g, and energy consumption from 0.007 kWh to 0.876 kWh across datasets. This positions LFD as the most resource-demanding among distributional representations. It must be emphasised that to generate LFDs, we use the Transformers variant of their respective spaCy pipelines, which contributes to the higher computational costs. It may be possible to reduce the costs of this process by using less computationally expensive models, however, this could potentially impact performance. Out

Table 8.2: Average cost of representation construction (and associated analyses) to predict for a single target domain, including the cost of building representations for associated intermediate tasks. RT indicates the wall-clock time for each representation in minutes. CO_2eq details the CO_2 equivalents (in grams) for constructing representations, calculated using the CodeCarbon [55] package. kWh shows the energy consumption in kilowatt-hours. These metrics are contrasted with the Sentence-BERT (**SEmb**) baseline and the average cost of training intermediate $\mathcal{D}_I$ models for the same domain. All costs account for base construction, divergence computation, and additional representation-specific costs.

	RT (m)				CO_2eq (g)				kWh			
	AM	YL	YH	IS	AM	YL	YH	IS	AM	YL	YH	IS
Distributional												
TD	7.69	4.35	1.97	0.24	14.32	8.07	3.63	0.45	0.053	0.030	0.014	0.002
LFD	99.58	55.66	27.67	0.72	235.00	132.34	66.45	1.81	0.876	0.493	0.248	0.007
KFD	4.48	2.3	1.16	0.09	10.60	5.73	3.17	0.19	0.040	0.021	0.012	0.001
Embedding												
BERT (C)	86.43	31.84	18.91	1.15	191.78	70.73	42.21	2.77	0.715	0.264	0.157	0.010
PWE	37.11	11.63	6.59	0.49	65.92	21.07	11.89	0.90	0.246	0.079	0.044	0.003
DCE	89.79	31.11	18.13	1.65	189.53	66.54	38.90	3.53	0.706	0.248	0.145	0.013
TOTAL	325.08	136.89	74.43	4.34	707.15	304.48	166.25	9.65	2.636	1.135	0.620	0.036
Baselines												
SEmb	22.26	10.17	4.91	0.80	52.30	24.07	12.33	1.88	0.195	0.090	0.046	0.007
Model Training												
$\mathcal{D}_I$	692.98	237.39	119.20	48.45	1346.30	468.46	236.49	96.82	6.442	2.241	1.132	0.463

of our embedding representations, Contextual BERT Embeddings (BERT (C)) in representations are the most resource-intensive, with time up to 86.43 minutes, CO_2 emissions up to 191.78g, and energy use up to 0.715 kWh.

- **Term Distributions (TD) are very efficient**. TD emerges as the most efficient amongst distributional representations, with minimal resource usage (time as low as 0.24 minutes, CO_2 emissions as low as 0.45g, and energy consumption as low as 0.002 kWh), considerably lower than LFD.

- **Sentence-BERT (SEmb) is the cheapest embedding representation, followed by Probability-Weighted Embeddings (PWE)**. SEmb shows the lowest resource demand amongst embeddings (time up to 22.26 minutes, CO_2 emissions up to 52.30g, energy use up to 0.195 kWh), followed by PWE (time up to 37.11 minutes, CO_2 emissions up to 65.92g, energy use up to 0.246 kWh). Both are significantly more efficient than DCE.

- **Resource-intensive representations are not necessarily more effective.** Despite their higher resource consumption, LFD and DCE do not always outperform

the more efficient TD and PWE in task selection, as shown in the previous section. This finding emphasises that higher resource usage does not equate to better performance.

- **Predicting task selection is cheaper than a grid-search of models**. The training of intermediate models ($\mathcal{D}_I$) is notably resource-intensive, consuming substantial time (up to 692.98 minutes), CO_2 emissions (up to 1346.30g), and energy (up to 6.442 kWh) across datasets. This dwarfs the resource usage of individual representations, highlighting the significant additional cost associated with model training.

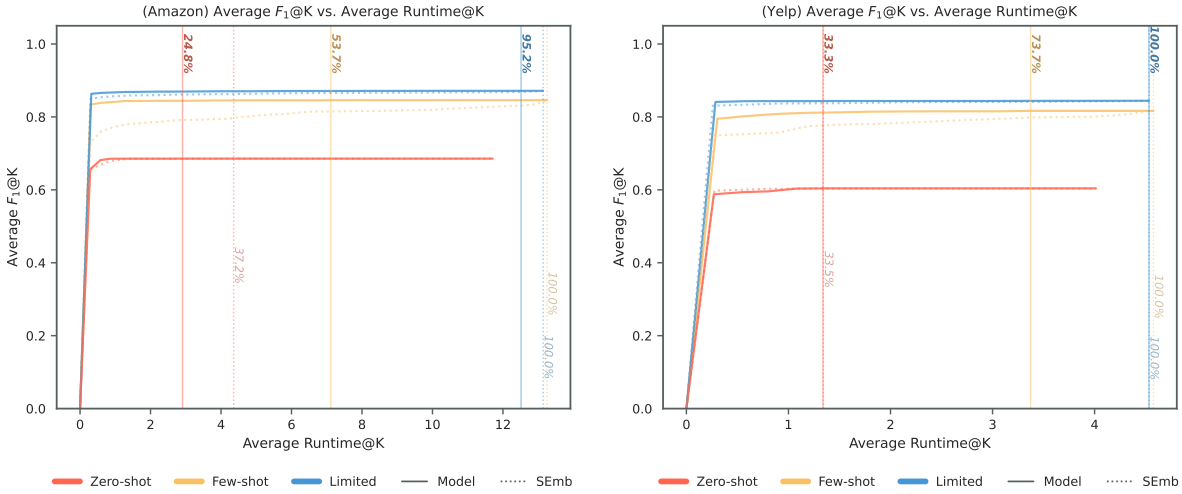
Our analysis shows clear differences in the resources needed to build and analyse different representations. Amongst distributional representations, Term Distributions (TD) are the most resource-efficient, using far fewer resources than Linguistic Feature Distributions (LFD). For example, for the Amazon dataset, TD needs only 7.69 minutes of time, 14.32 grams of CO_2, and 0.053 kWh of energy, compared to the much higher demands of LFD. Among embedding representations, Sentence-BERT (SEmb) and Probability-Weighted Embeddings (PWE) are cost-effective. SEmb is particularly resource-efficient, while PWE provides a good balance of performance and resource use. When compared to the extensive resources required for training intermediate models ($\mathcal{D}_I$), the efficiency of TD, SEmb, and PWE is even more evident. In summary, we find significant differences in runtime, CO_2 emissions, and energy consumption across representations, answering RQ4.

8.6 Performance-Efficiency Analysis

In this experiment, we evaluate the balance between the performance and efficiency of our task selection approach. The aim is to demonstrate the time and resource savings achieved by our method compared to both a full grid search of all task combinations and the baseline method (Sentence-BERT, SEmb). We measure efficiency in terms of the cumulative maximum F_1-score against the cumulative runtime (RT) for each model up to a given rank (K). Based on the performance of our representations (see Section 8.4.2), we anticipate that we will be able to find the best models for each dataset quickly, particularly in zero-shot settings. Based on the same results, in few-shot and limited settings, we expect to beat the SEmb baseline.

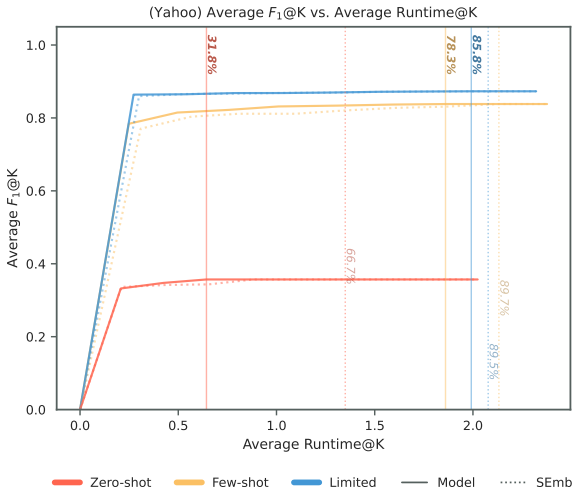
Figure 8.5 contrasts the performance of our task selection approach against both a full grid-search over all tasks and a Sentence-BERT (SEmb) baseline across different datasets. It plots the Average $F1@K$ values against Average Runtime@K values, where averaging is performed across each domain within each dataset. Solid lines represent the performance vs. runtime of our task selection method, while dotted lines represent the SEmb baseline.

Figure 8.5: Average F_1@K values plotted against Average Runtime@K, aggregated across target domains. Solid lines denote the performance vs. runtime of our task selection approach, dotted lines denote the Sentence-BERT (SEmb) baseline. Red, yellow, and blue lines denote zero-shot, few-shot, and limited experimental settings, respectively. Vertical lines in matching colours, pinpointing the moment where the model achieves the maximum F_1-score for each experimental setting; annotations on these lines indicate the specific percentage of total model runtime at which the maximal F_1 is reached.

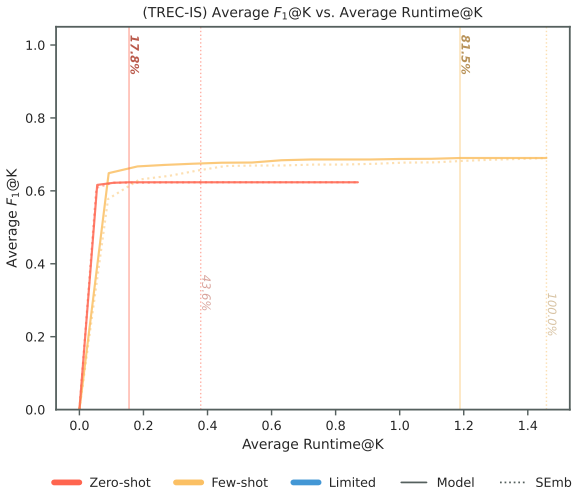


(a) F_1@K vs. Runtime@K, Amazon.

(b) F_1@K vs. Runtime@K, Yelp.



(c) F_1@K vs. Runtime@K, Yahoo.



(d) F_1@K vs. Runtime@K, TREC-IS.

Red, yellow, and blue lines correspond to zero-shot, few-shot, and limited experimental settings, respectively. Vertical lines in matching colours (and varying styles) indicate the point of maximum F_1-score for each experimental setting whereas annotations on these lines show the specific percentage of total model runtime at which the maximum F_1 is achieved. Key observations include:

- **There are significant runtime reductions in zero-shot settings**. Our ap-

proach notably reduces runtime in the zero-shot setting across all datasets, with Amazon seeing a reduction from 11.7 hours to 2.9 hours (75.2% reduction), and Yelp from 4.01 hours to 1.33 hours (66.6% reduction). In the Yahoo and TREC-IS datasets, runtime reductions are 68.2% and 82.2%, respectively.

- **Few-shot settings show varied gains in efficiency**. The few-shot setting on Amazon shows a 46.3% runtime reduction, while Yelp sees a 26.3% decrease. Yahoo and TREC-IS reports reductions of only 21.7% and 18.5%.

- **Gains in limited settings are minimal**. In the limited setting, Yahoo shows a 14.2% reduction in runtime while Amazon only reports a 4.8% reduction in runtime. Yelp's efficiency matches that of a grid search, underscoring the difficulty in predicting transfer gain in these experimental settings.

- **We save more runtime vs. the SEmb baseline**. When compared to the SEmb baseline, we save an extra 1.5 hours in runtime (2.9 vs. 4.4 hours runtime) for the Amazon dataset in zero-shot settings, amounting to 0.17kg in CO_2 emissions and 0.81 kWh. We save the same amount in few-shot and limited settings as with a full grid search of tasks, where SEmb does not provide any efficiency gains. For Yelp, our approach is as good as the SEmb baseline in zero-shot settings. For Yahoo and TREC-IS, we save an extra 34.9% (0.24 hours) and 25.8% (0.22 hours) in the zero-shot setting, respectively. In the Yahoo dataset, for few-shot and limited settings, SEmb manages to achieve 10.3% and 10.5% reductions when compared to a full grid-search. Our approach remains more efficient, in terms of model runtime, than the baseline here, saving an extra 11.2% in few-shot and a marginal 3.7% in limited settings when compared to SEmb.

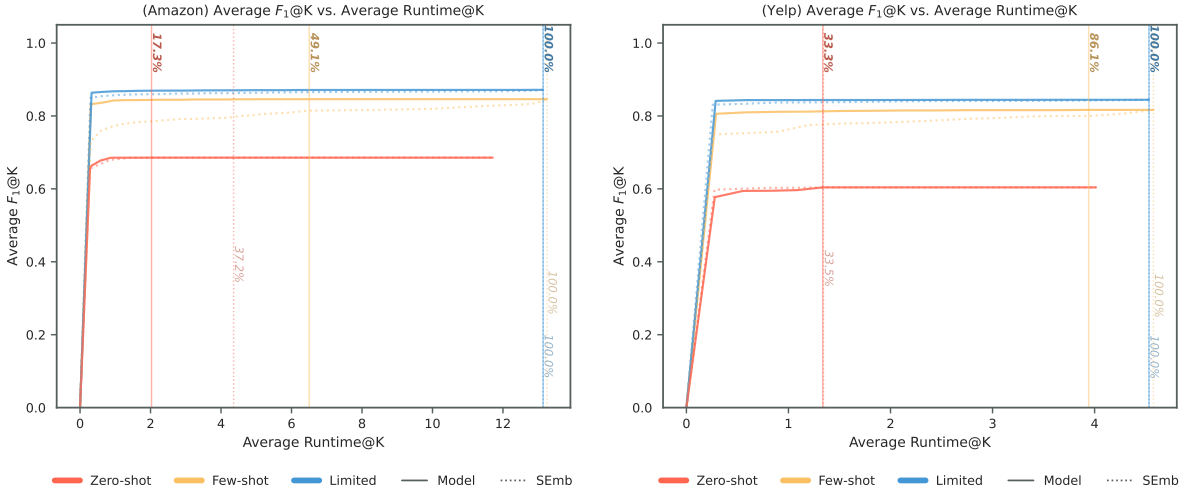
Overall, we save 22.7 hours, 2.61kg CO_2, and 12.536 kWh compared to a grid search. When compared to the SEmb baseline, we save 13.28 hours of runtime, 1.54 CO_2, and 7.443 kWh. Accounting for the costs of creating these representations and excluding *SEmb* (Table 8.2)—9.01 hours, 1.19kg CO_2, and 4.427 kWh—the net savings amount to 13.69 hours, 1.42kg CO_2, and 8.109 kWh for a grid search. However, when comparing to the *SEmb* baseline, the reductions are notably smaller: 4.27 hours, 0.35kg CO_2, and 3.016 kWh. This observation prompts further investigation through ablation studies, where the removal of the most resource-intensive representations is analysed to assess its impact on efficiency.

8.6.1 Performance-Efficiency: Ablation Study

In this ablation study, we examine the effects of removing Linguistic Feature Distributions (LFD) and Distributive Contextual Embeddings (DCE) from our feature set due to their

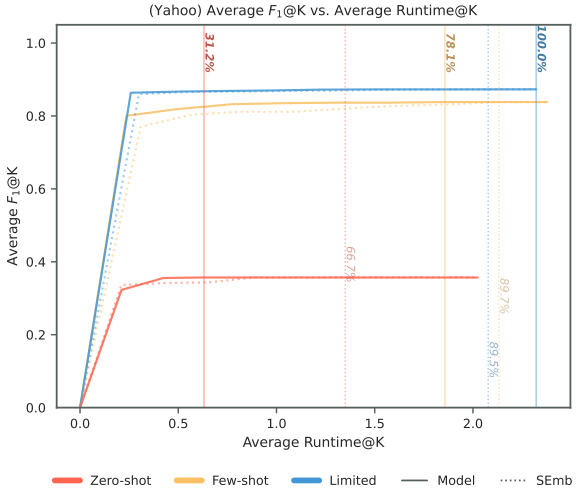
high resource-intensiveness. Given its competitive performance and low associated costs, particularly in zero-shot settings, we also incorporate Sentence Embeddings (SEmb) into our feature set. The goal of this experiment is to assess the trade-offs between performance and computational resource use. Here, we anticipate some loss in performance relative to using the entire feature set from the previous experiment.

Figure 8.6: Average F_1@K values plotted against Average Runtime@K, aggregated across target domains after the removal of LFD/DCE features and the inclusion of SEmb as a feature. Solid lines denote the performance vs. runtime of our task selection approach, dotted lines denote a SEmb baseline. Red, yellow, and blue lines denote zero-shot, few-shot, and limited experimental settings, respectively. Vertical lines in matching colours, pinpointing the moment where the model achieves the maximum F_1-score for each experimental setting; annotations on these lines indicate the specific percentage of total model runtime at which the maximal F_1 is reached.

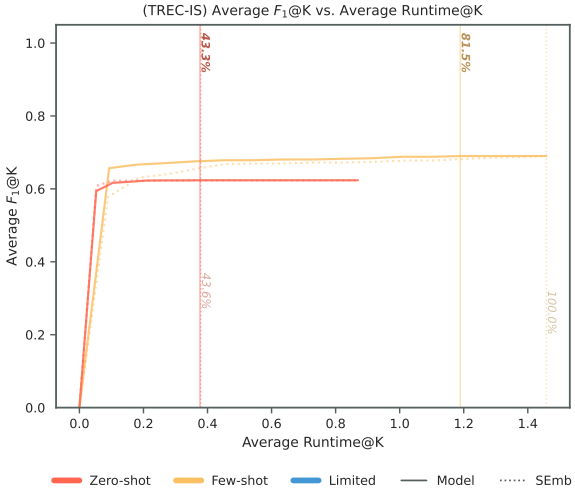


(a) F_1@K vs. Runtime@K, Amazon.

(b) F_1@K vs. Runtime@K, Yelp.



(c) F_1@K vs. Runtime@K, Yahoo.



(d) F_1@K vs. Runtime@K, TREC-IS.

Figure 8.6 plots the Average F_1@K values against Average Runtime@K, aggregated

across target domains, similar to our previous analysis, but after the removal of LFD/DCE features and the inclusion of SEmb. Key observations include:

- **Removing LFD/DCE negatively impacts runtime efficiency**. The removal of LFD and DCE affects the ability to reduce runtime in limited settings for Amazon and Yahoo datasets, suggesting LFD or DCE's influence in previous reductions.
- **Additional savings in zero-shot settings, but with trade-offs**. In the zero-shot setting, we save an additional 7.5% in runtime for Amazon, but our time taken to find the maximum performance for TREC-IS increases from 17.8% to 43.3%, at a cost of 0.22 hours, 0.03kg CO_2, and 0.128 kWh (along with a marginal 0.6% saving for Yahoo), indicating a trade-off in efficiency and performance across different datasets. After the ablation studies, our approach yields a net saving of 0.67 hours, 0.07 kg CO_2, and 0.365 kWh across all datasets and settings.
- **Mixed results in the few-shot setting**. After removing LFD/DCE (and adding SEmb), we save an additional 4.6% in the Amazon dataset (0.61 hours, 0.07kg CO_2, 0.355 kWh). However, for Yelp, we see a 12.4% increase in runtime (0.57 hours, 0.07kg CO_2, 0.327 kWh). In total, the effects of this ablation study result in a marginal net improvement of 0.04 hours, equivalent CO_2, and 0.03 kWh saved.
- **Increased costs in the limited setting**. Compared with Figure 8.5, there is an increase of 3.24 hours, 0.38 kg CO_2, and 1.822 kWh when compared to using the full representation set. This underscores the value of the removed representations (LFD/DCE) for reducing costs in limited settings, despite the costs incurred to produce and evaluate these representations.

When considering model savings alone, these results translate to a loss of 2.53 hours, an additional 0.31 kg of CO_2, and 1.427 kWh in total. However, the total expenditure in representation costs after the removal of LFD and DCE (and including SEmb) drops to 4.24 hours, 0.544 kg CO_2, and 2.029 kWh. This adjustment leads to notable gains across different settings: for Amazon in the zero-shot setting, there's a saving of 5.65 hours, 0.746 kg CO_2, and 2.885 kWh; in TREC-IS zero-shot, the savings are 4.55 hours, 0.616 kg CO_2, and 2.27 kWh. In the few-shot domain for Amazon, the improvement is 5.38 hours, a reduction of 0.716 kg CO_2, and 2.753 kWh less energy usage, and for Yelp, the savings are 4.2 hours, 0.576 kg CO_2, and 2.071 kWh. This means that, in total, our method effectively saves an average of 2.24 hours, 0.266 kg CO_2, and 2.018 kWh, offsetting the costs incurred by reduced effectiveness of task selection itself. Overall, the ablation studies highlight the trade-offs between computational efficiency and performance. While the removal of resource-intensive representations results in reduced effectiveness in the task selection process itself, the overall picture is one of clear net benefit, highlight the

need for careful considerations in resource-efficiency across different components of the framework.

In response to RQ5, our comprehensive analysis reveals a clear advantage of our task selection approach in achieving maximal F_1-score performance efficiently. Using the full feature set, we observed substantial savings of 22.7 hours, 2.61kg CO_2, and 12.536 kWh compared to a grid search. Even against the SEmb baseline, our approach saved 13.28 hours of runtime, 1.54 kg CO_2, and 7.443 kWh. These outcomes underscore the efficiency of our method over traditional grid searches and established baselines. Furthermore, our ablation studies, which involved removing resource-heavy representations (and including SEmb), further highlight the effectiveness of this approach. Despite some increases in model runtimes, the net resource savings (in addition to the savings using the full feature set)—2.24 hours, 0.266 kg CO_2, and 2.018 kWh—demonstrate that our method balances performance well with computational efficiency.

8.6.2 Maximal Performance vs. Runtime Analysis

Figure 8.7: Scatter plot illustrating the trade-off between maximum F_1-score and model runtime savings compared to a full grid search for each domain. Different colours describe the datasets: red for Amazon, yellow for Yelp, blue for Yahoo, and green for TREC-IS. Vertical lines and their annotations in the same hues show the average maximum runtime for all intermediate domains, for each dataset. Marker shapes denote the experimental setting: crosses for zero-shot, plus signs for few-shot, and circles for limited.

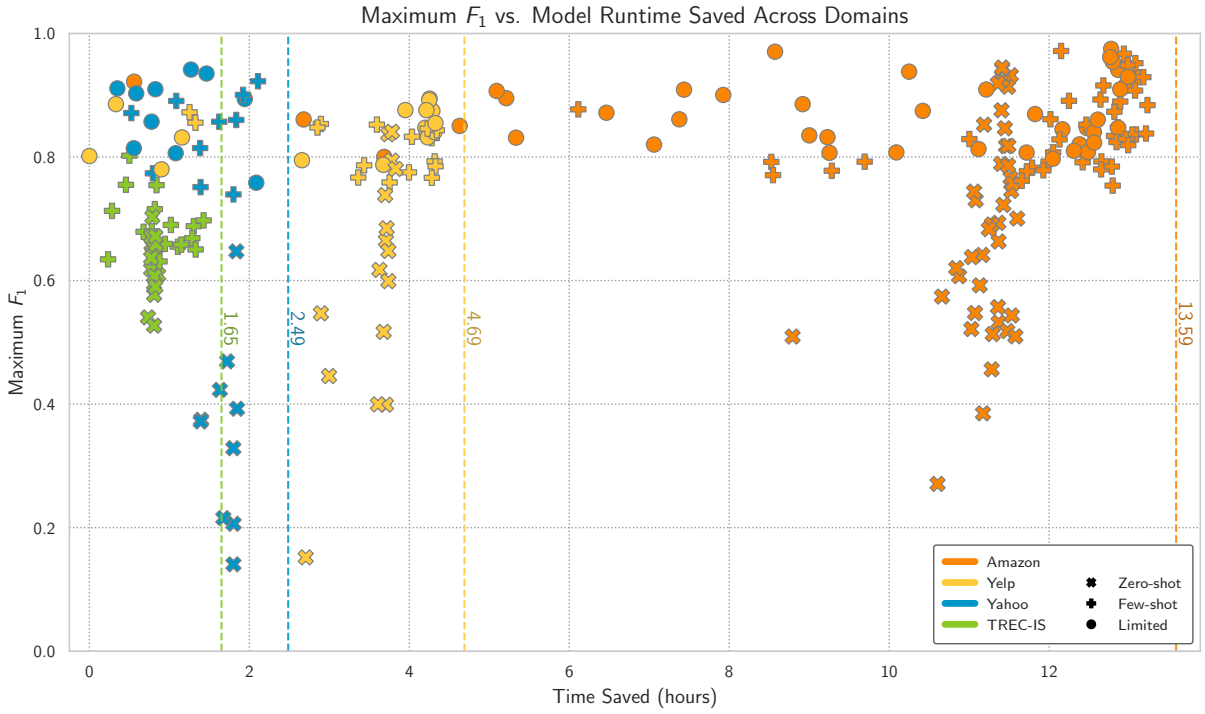


Figure 8.7 demonstrates the relationship between the maximum F_1-score and model run-

time savings, using a scatter plot aggregated across different domains. The plot employs colour codes and marker shapes to differentiate datasets and experimental settings: red, yellow, blue, and green for Amazon, Yelp, Yahoo, and TREC-IS, respectively; and crosses, plus signs, and circles for Zero-shot, Few-shot, and Limited settings, respectively. To highlight the savings relative to the maximal performance gained for each domain, we present the following findings:

- **Amazon**. For Amazon, the total runtimes are 11.7, 13.25, and 13.14 hours for zero-shot, few-shot, and limited settings, respectively. Remarkably, the zero-shot setting for Amazon demonstrates substantial efficiency, saving about 11 hours of model runtime on average, which translates to a 96.5% reduction in runtime. In contrast, the few-shot and limited settings show a lesser yet significant reduction, averaging 92% and 72.9%, respectively. When factoring in the representation cost, which amounts to 5.79 hours for all representations, the savings are adjusted to 47% for zero-shot, 48% for few-shot, and 28% for limited settings.

- **Yelp**. Yelp's dataset shows runtimes of 4.01, 4.58, and 4.53 hours for the three settings. Here, the average savings are approximately 90% for zero-shot and around 78% for both few-shot and limited settings. After incorporating the representation cost (2.5 hours), the savings are still notable, standing at 28.9%, 24.7%, and 24.5% for zero-shot, few-shot, and limited settings, respectively.

- **Yahoo**. For Yahoo, the total runtimes are shorter, with 2.02, 2.38, and 2.32 hours for zero-shot, few-shot, and limited settings. The model runtime savings here are between 1.49 and 1.73 hours, translating to 64.2%, 67.2%, and 85.6% for the respective settings. Factoring in the representation cost (1.32 hours), the savings are 20.2%, 11.7%, and 7.2%.

- **TREC-IS**. TREC-IS dataset, with runtimes of 0.87 and 1.46 hours for zero- and few-shot settings, shows significant efficiency in zero-shot settings, saving about 0.81 hours (93.1%). The few-shot setting saves less, at 70%. After accounting for the representation cost (5.14 minutes or 0.085 hours), the savings stand at 83.3% and 64% for zero-shot and few-shot, respectively.

Our observations effectively demonstrate that the proposed framework exhibits high efficiency in task selection for transfer learning across various datasets, while using the full breadth of our representation set. Adjusting for the cost of representation construction and evaluation using this set, the most pronounced savings are observed in the zero-shot settings, with between 20.2% and 83.3% saved in this setting. Even in more difficult experimental settings, we save between 11.7-67.2% and 7.2-28% for few-shot and limited

settings. This concludes our final experiment, demonstrating that the maximal performance of each domain is found earlier in the task selection process compared with a full grid-search of task combinations, thereby answering our sixth research question.

8.7 Conclusions

In this chapter, we conclude our experimentation through the investigation of the Ranking Intermediate Tasks component of our framework. We began by outlining methodologies for predicting relative transfer gain using the representations and their best settings that were identified in previous chapters (Section 6.8 and 7.8). Our approach consisted of using regression and ranking models to predict the ranking of intermediate models for transfer according to the inherent ordering amongst their performance scores. Using these models, we demonstrated that our method was very effective in ranking tasks for transfer. Furthermore, we demonstrated the practical benefits of this approach, contrasting the performance of models across domains with the time taken by our models to find them.

In particular, in Section 8.2, we established an approach for task selection which was centred around evaluating the quality of predicted rankings of task pairs.Subsequently, we outlined the procedure for evaluating the system using three key criteria: *Performance*, *Efficiency*, and the balance between Performance and Efficiency. Section 8.2.4 described our strategy for evaluating performance which included establishing baseline performances, evaluating performance in terms of the frequency of optimal models found within specific rank intervals, and analysing the performance of each representation category individually. In Section 8.2.5, we defined our process for evaluating the efficiency of representation construction and evaluation. Lastly, in Section 8.2.6, we discussed our approach to evaluating the trade-offs between performance and efficiency through the practical application of our framework.

In Section 8.3, we defined the research questions that we investigated in this chapter, dividing our approach into six individual experiments. The first three focused on evaluating the performance of the system, the third on the efficiency of representation construction and evaluation, and the remaining two on the considerations in balancing these two criteria. The first set of experiments involved comparing the performance of our task selection system against a random ordering of intermediate models and against an established baseline, Sentence-BERT Embeddings (SEmb). Following this, we evaluated how often our model found the optimal intermediate model for each domain, relative to the size of each dataset. Lastly, we analysed the performance of each representation category individually, according to its NDCG at different ranks. In the second set of experiments, we considered the cost incurred by an end user of such a framework by reporting the average cost of computing and evaluating the representations required to use the framework

to predict the ranking of tasks for a single domain. Here, we compared against the cost of computing SEmb and of training each intermediate model. In the final set of experiments, devised a framework for evaluating the time taken to find the maximum performance across all domains in each dataset, formulating task selection as a "search" task from the perspective of a user, where as the user traverses down a list of predicted (ranked) models, the likelihood of them finding a better model increases at the cost of end-to-end runtime. We expanded upon this analysis by aiming to improve its efficiency, performing an ablation study to remove the most expensive representations as input features to the model and measuring the trade-offs between performance and efficiency. Lastly, we evaluated the average savings achieved by our models to find the maximal performance for each domain.

Through the experiments detailed in Sections 8.4, 8.5, and 8.6, we have systematically evaluated our transferability estimation framework. We found that our framework rapidly identified optimal models in zero-shot settings. However, in few-shot and limited data scenarios, while still effective, the performance varied more broadly. These findings provide an avenue for future work in estimating transferability in more difficult settings, as the accuracy of our method decreased in these more constrained situations. Our analysis showed that the costs incurred by different representations varied greatly. Term Distributions were very cost-effective, while SEmb and Probability-Weighted Embeddings (PWE) struck a good balance between performance and cost. The results from these experiments inform future work in efficient representation construction. In evaluating performance and efficiency trade-offs, our approach saved a lot of time and resources while still performing well, especially when compared to a full grid-search of models and established baselines. Following our ablation study, our results emphasised the need to consider efficiency in each component of the framework: removing expensive representations incurred less costs earlier on but negatively impacted the effectiveness of task selection itself. Nevertheless, we demonstrated how significant savings in representation costs could offset the reduced effectiveness of the algorithm.

The experiments and analyses in this chapter directly address our second thesis research question (RQ2 in Section 1.3), which focuses on the practical implications of our proposed framework. By comparing our approach to grid search methods and strong baselines, we demonstrate that our framework substantially reduces the time and resources required for task selection while maintaining performance. The evaluation of performance, efficiency, and their trade-offs provides a comprehensive assessment of our approach. The significant cost savings achieved by our models in obtaining optimal transfer performance compared to exhaustive search highlight the practical value of our framework. The findings from each of our experiments results in the main contribution of this thesis: to produce a comprehensive and efficient framework for transferability estimation in natural language processing.

Chapter 9

Conclusions and Future Work

9.1 Contributions and Conclusions

This thesis hypothesised that by constructing and evaluating effective domain representations using statistical divergence measures, we can significantly improve the accuracy of intermediate task selection in transfer learning, and more importantly, that the magnitude of this improvement is predictable. During the course of this thesis, we have proposed a novel, end-to-end framework that defines five components, which enable us to estimate the success of transfer learning for natural language processing tasks. Through the construction and evaluation of distributional and embedding representations, we conclude that we can, indeed, significantly improve the accuracy of task selection across a broad range of textual domains. Moreover, we also argued that a systematic, divergence-based approach to task selection will substantially reduce the time and resources required by exhaustive grid search methods. Following our experiments in Chapters 6-8, we conclude that we can, indeed, significantly reduce the time taken to find the most optimal models for transfer. The remainder of this section discusses the contributions and conclusions of this thesis in more detail.

9.1.1 Contributions

The main contributions of this thesis are as follows:

- In Chapter 3, we proposed a novel framework for estimating transferability for natural language processing tasks. This framework defines five components required for ranking and recommending optimal tasks for transfer. In particular, the Domain Adapter Generation and Domain Transfer Analysis components (see Section 3.4) involve the training and evaluation of the intermediate, target, and intermediate-to-target models. The Representation Construction and Divergence Estimation components (see Sections 3.5 and 3.6) involve the construction and evaluation of

representations based on their correlation with relative transfer gain. Finally, the Intermediate Task Selection component 3.7 component integrates task divergence estimations and intermediate-to-target model evaluations into a method of selecting intermediate tasks for transfer. In Section 1.1, the challenges presented in estimating transferability for language-based tasks were motivated. This framework is a key contribution of this thesis, in that it provides a systematic, modular approach to selecting tasks for transfer learning through a thorough analysis of both established and novel methods of representing textual domain data. To the best of our knowledge, there have been no previous studies investigating task selection in this detail.

- In Chapter 4, we presented a comprehensive taxonomy of statistical divergence and diversity measures. These methods enabled us to evaluate the similarity between of distributional and embedding representations and to quantify the characteristics of individual representations, providing useful features for our task selection models. In particular, we defined the geometric, information-theoretic, optimal transport, statistical moments, and diversity measures used throughout this work. We build upon a taxonomy from prior work [46], extending it with additional measures. This contribution is important in formalising an approach to using domain divergence as a proxy for transferability estimation.

- In Chapter 5, we presented the diverse datasets—and the strategies to transform them into topic classification tasks to maintain consistency across datasets—that were used to validate our thesis statement. In particular, we transformed the Amazon Product Reviews dataset from a rating prediction task into a topic classification task using its metadata. Similarly, we merged and restructured data from the 2015 Yelp Dataset Challenge to create the Multi-Domain Yelp Business Reviews dataset for topic classification. Additionally, we used the Yahoo! Answers, a question and answer topic classification dataset, and TREC Incident Streams datasets, a crisis classification dataset, to diversify our analysis. A unique aspect of our approach was dividing these domains into 85 binary classification tasks using a One-Vs-Rest strategy, enabling a large-scale assessment of transferability estimation. We also discussed how we efficiently train and store the models produced and evaluated in the thesis, including the use of adapter modules to limit trainable layers, thereby mitigating catastrophic forgetting, and our use of mixed precision training. We concluded the chapter by demonstrating that the gains/losses of randomly selecting intermediate tasks carried significant risk of performance degradation, motivating the need for a transferability estimation framework.

- In Chapter 6, we explored term, linguistic feature, and topic frequency-based dis-

tributional representations and evaluated them based on how well they correlated with transfer learning success. We adjusted specific features of these representations to see how they affected their performance. Additionally, we introduced methods to measure divergence at different levels: instance, centroid, and domain. The primary contribution of this chapter is the empirical validation of a key hypothesis underlying our thesis: by constructing effective distributional representations, we have demonstrated a strong relationship between distributional representation divergence and relative transfer gain, especially in zero-shot settings. This finding not only confirms the rationale behind our approach but also substantiates the effectiveness of our proposed methodologies.

- In Chapter 7, we investigated the use of embedding representations. To the best of our knowledge, the scope and extent of this investigation is not found in prior literature. We combined information from frequency-based distributions to weight cheap, static, embeddings, finding comparable performance to more resource-intensive contextual embeddings. Furthermore, we introduced a novel approach to comparing variation in term contexts, introducing a representation that combines the granularity of term distributions with the expressiveness of embeddings. Hence, the main contributions of this chapter were: (1) the enhancement of term vectors by incorporating term frequency distributions, achieving results that, in 5 out of 8 settings (that reported at least one significant value), surpassed more resource-intensive contextual embeddings; and (2) the introduction of a new method to compare aggregated term vector distributions, Distributive Contextual Embeddings, effectively measuring how term context varies across domains.

- Finally, in Chapter 8, we presented our divergence-based approach to intermediate task selection. We thoroughly examined the effectiveness of our framework through ranking and transferability metrics, comparing the performance of each representation individually and using all representations. We also leveraged the resource-tracking (see Section 3.3) component of our framework to provide a comprehensive, cost-based breakdown in terms of end-to-end runtime, CO_2 emissions, and energy expenditure. Lastly, we compared the practical application of our approach, considering how an end-user would use our framework to find optimal intermediate tasks quickly. This chapter's contributions can be summarised as: (1) up to an 83.3% runtime saving in identifying the most optimal intermediate domains for transfer learning; (2) a comprehensive breakdown of the costs incurred in constructing and evaluating each of our representations in terms of end-to-end runtime, CO_2 emissions, and energy usage; (3) we save 22.7 hours, 2.61kg CO_2 emissions, and 12.536 kWh compared to an exhaustive grid search of all task combinations.

9.1.2 Conclusions

The main achievements and conclusions of this work are as follows:

Effectiveness of Distributional Representations. From the experiments detailed in Chapter 6, we conclude the following. Distributional Representations, in zero-shot scenarios, result in strong to very strong ($0.4 \leq \rho < 0.7$ and ≥ 0.7, respectively, in terms of Spearman's ranking correlation coefficient, ρ) correlations with performance. This was particularly evident in our evaluations of individual distributional representations in Section 8.2. However, distributional representations had difficulties in predicting transfer gain in experiment settings when models had undergone further fine-tuning on target domain data (few-shot and limited settings). This prompted further investigation into the calculation of divergence at different levels of abstraction where, with the exception of Linguistic Feature Distributions, we found that distributional representations were no more effective at more granular (instance- and centroid-based) levels of comparison when compared to domain-aggregated baselines. For representation-specific configurations, we found that: (1) constructing Term Distributions (TD) from a target-focused vocabulary yielded the strongest correlations; (2) Linguistic Feature Distributions (LFD) reported strong correlations in zero-shot settings, but performed poorly in few-shot and limited settings; (3) Topic Frequency Distributions (KFD) resulted in comparatively weaker correlations with relative transfer gain than other distribution types. From further, representation-specific analysis in Section 8.4, we found that TD yielded the best performance in intermediate task ranking amongst all distributional representations while being very cheap to produce, often surpassing more resource-intensive methods.

At the beginning of this thesis, we asked whether advanced domain representation and precise divergence estimation techniques can lead to more effective estimations of transfer learning success. From the experiments in Chapter 6, we have shown that this is indeed the case, however, while distributional representations demonstrated a strong relationship with relative transfer gain in zero-shot settings, they performed comparatively poorly in more difficult experimental settings, prompting the investigation into the use of more complex, embedding representations.

Effectiveness of Embedding Representations. From the experiments in Chapter 7, we conclude the following. Embedding representations, unsurprisingly, often reported very strong correlations with transfer gain/loss. In particular, embedding representations often surpassed distributional representations in their ability to capture domain characteristics in difficult settings, particularly evidenced by their performance in representation-specific evaluations in Section 8.4 for few-shot and limited settings. Amongst embedding representations, we were able to improve the quality of cheap, general-purpose embeddings through frequency-based information extracted from term distributions. These weighted embed-

dings often matched or surpassed the performance of significantly more resource-intensive contextual BERT embeddings, offering a cheaper, highly expressive representation at a fraction of the cost. A core contribution of Chapter 7 was the introduction of Distributive Contextual Embeddings (DCE). Combining concepts from both distributional and embedding-based approaches, our approach to evaluating the diversity and variance between term vector distances resulted in the only representation to provide strong to very strong correlations with transfer gain in both few-shot and limited settings. Indeed, further experiments in Section 8.6 highlighted the importance of DCEs in contributing to a reduction in runtime in difficult experimental settings.

This thesis postulated that we could measure a direct relationship between the domain divergence of representations and transfer learning success. We found that, embedding representations improved upon the performance of distributional representations from the experiments in Chapter 6, particularly in few-shot and limited settings.

Accuracy and Efficiency of Intermediate Task Selection. In our final chapter, we provided a comprehensive summary of the performance and efficiency of our task selection approach, highlighting the practical application of our transferability predictors. We provided a thorough examination of performance in terms of: (1) comparing our overall method to established baselines; (2) quantifying the proportion of optimal intermediate models found in the top rankings of our predicted outputs; and (3) comparing the contribution of different representation types to effectively predict the ranking of intermediate tasks. Through our analyses, we found that: our approach outperformed baseline methods, often reporting > 0.9 NDCG, that a regression approach was more effective in zero-shot scenarios, and that ranking approaches excelled in difficult few-shot and limited settings. Through an analysis of the top predicted ranks in the zero-shot setting, we found that we were able to find optimal models quickly, where all of the optimal intermediate models were identified within the top-5 ranks for Yelp, Yahoo, and TREC-IS datasets, and all but one identified in these intervals for Amazon. In few-shot and limited settings, unsurprisingly, we encountered increased difficulty in identifying optimal tasks, where the distribution of optimal models were more spread across the ranks. Nevertheless, we managed to find between 41-85% and 47-68% of optimal intermediate models in the top-5 ranks for few-shot and limited settings, respectively. In our efficiency analysis, we were able to identify the most resource-intensive representations—in terms of runtime costs, CO_2 emissions, and kWh used—LFDs and DCEs for distributional and embedding representations, respectively. These findings informed our ablation studies in Section 8.6.1 studies, in which we found that removing the most expensive representations—LFDs and DCEs—had an overall negative impact on the accuracy of task selection, but that the costs saved by removing these representations offset the drop in accuracy. Finally, we summarised the overall performance-efficiency trade-offs of our approach, using all features. We compared

the average costs saved relative to finding the optimal intermediate task for each domain, where we concluded—after adjusting for the cost of constructing our representations—the following: (1) we were able to save between 28-47% in end-to-end runtime for the Amazon dataset; (2) we reduced the time taken to find the best task combinations in the Yelp dataset by between 24.5-28.9%; (3) our total, average cost savings per domain for the Yahoo dataset was between 7.2% and 20.2%; and (4) the largest range of cost savings were found in the TREC-IS dataset, where we saved between 64-83.3% in end-to-end runtime.

In our thesis statement (see Section 1.2), we hypothesised that constructing and evaluating effective domain representations using statistical divergence measures can significantly improve the accuracy of intermediate task selection in transfer learning, and, more importantly, that the magnitude of this improvement is predictable. Based upon our performance experiments in Chapter 8, we conclude that a systematic approach to task selection, through the evaluation of domain divergence between effective representations, can significantly improve the accuracy of task selection.

We further argued that a systematic, divergence-based approach to task selection will substantially reduce the time and resources required by exhaustive grid search methods. Based upon our performance-efficiency experiments, we conclude that our approach significantly reduces the resources and time required by exhaustive grid search methods. This, in turn, results in more accurate estimations of transferability and facilitates better model development in transfer learning.

9.1.3 Thesis Conclusions

This thesis aimed to address two key research questions (RQs):

RQ1. How can we effectively construct and evaluate domain representations to capture the characteristics and relationships between different domains, and to what extent can statistical divergence measures, applied to these representations, accurately estimate the transferability between tasks and improve the selection of optimal intermediate tasks for transfer learning?

RQ2. How does a systematic, divergence-based approach to task selection compare to exhaustive grid search methods in terms of both accuracy and efficiency (time and computational resources), and how can we quantify and analyse the trade-offs between performance and efficiency to guide the development of more effective and sustainable transfer learning practices?

We proposed a novel framework for transferability estimation, comprised of key components, and argued that a systematic approach to task selection could significantly reduce resources and time. In Chapters 6-8, we empirically investigated our hypotheses and concluded the following:

- Addressing **RQ1**, we demonstrated that advanced construction of distributional and embedding representations proved effective in encapsulating domain characteristics. Our novel Distributive Contextual Embedding approach, combining distributional data with embeddings, significantly enhanced domain representation quality (see findings in Chapter 6 and Chapter 7). These findings support our hypothesis that improved domain representations and precise divergence estimations can improve the accuracy of transferability estimation for transfer learning in natural language processing.

- Addressing **RQ2**, our systematic approach to task selection, incorporating extensive domain transfer analysis across 85 domains, demonstrated high accuracy and significant efficiency improvements over both a grid search of all task combinations and competitive, established baselines. Our method consistently identified optimal intermediate tasks quickly, reducing time and computational resources substantially (see Section 8.6). These results confirm that a systematic, divergence-based approach to task selection can lead to more accurate, efficient, and sustainable transfer learning practices.

Hence, we conclude that by creating and comparing effective domain representations and accurately estimating domain divergence, a more efficient and systematic approach to task selection in transfer learning is achievable. This leads to not only time and resource savings but also enhances the overall effectiveness of model development for low-resource tasks in natural language processing.

9.2 Directions for Future Work

This section discusses several directions for future work related to the improvements of our framework:

- **Dataset Diversification**. In our study, by converting existing datasets into binary classification tasks, we limited our analysis to similar types of domains, primarily comparing within the same dataset. This limitation means we've only observed how models adapt to variations within a similar context. However, in real-world applications, the challenge often lies in transferring knowledge across vastly different domains—for example, applying insights from a healthcare dataset to a financial dataset. Future research should therefore include datasets from varied domains to understand how models can generalise across distinct datasets, focusing on cross-domain comparisons can provide a better understanding of the limitations and strengths of transfer learning when faced with fundamentally different types of data.

- **Generalisability to Other Domains and Tasks**: While this thesis focused on binary classification tasks across textual domains, future research could explore the generalisability of our findings to other types of tasks, such as multi-class classification or multi-task learning. Additionally, the principles of our framework could be extended to domains beyond text, such as speech, image, or video data, with appropriate modifications to the representation construction and divergence estimation techniques. However, adapting our methods to other contexts may involve challenges related to dataset availability, domain-specific preprocessing requirements, and computational resource constraints. Nonetheless, the potential benefits of our framework, including improved performance, reduced development time, and cost savings, make it a promising avenue for future research in a wide range of application areas.

- **Uncertainty Estimation**. In our Performance-Efficiency experiments, we simulate a scenario in which a user "searches" a predicted ranking of task pairs to try, where each step in the ranking improves the likelihood that the user will find a combination of tasks that will result in better performance, at the cost of additional runtime. The limitation in this approach is that, without access to ground truth performance scores, a user of such a framework would not know when the best task combination has been found. Hence, incorporating to estimate uncertainty in domain divergence and task selection is a potential area for growth. Future models can use Bayesian inference to provide not only predictions but also measures of the certainty of these predictions, providing an end-user with a measure of "confidence" that the best model combination has been found. This would require probabilistic models to assess the confidence of the framework in its decisions, leading to better informed decisions, in particular when dealing with uncertain or incomplete data.

- **Application to In-Context Learning Techniques**. Expanding on the findings of this thesis regarding the correlation between transfer learning performance and divergence in domain representations, future work could focus on designing in-context learning techniques that adaptively use information in a given target domain. For instance, research could develop methods to dynamically design within-prompt instructions based on the distributional characteristics of the language in the target domain. Such techniques could significantly enhance the precision of in-context learning by ensuring that the prompt instructions are optimally relevant to the task at hand, potentially reducing ambiguity and increasing the model's performance on target tasks.

- **Fostering an Open-Source Ecosystem for Transferability Estimation**. An avenue for future work is to evolve this framework into an open-source package. This

will make transferability estimation more accessible to researchers and developers. By offering our framework as an open-source tool, we aim to encourage wider participation and collaboration, enhancing its utility and applicability in various research contexts, fostering collaboration and innovation in the field.

9.3 Closing Remarks

In this thesis, we posited that advanced domain representation and precise divergence estimation could significantly enhance the effectiveness of transferability estimation in natural language processing. Through extensive research and empirical testing of our transferability estimation framework, we demonstrated that a systematic approach to task selection can indeed result in the accurate identification of optimal tasks and additionally, dramatically reduce the resources and time incurred through exhaustive grid search methods. Our investigations into representation construction, divergence estimation, and intermediate task selection have proven the effectiveness of our methods. In summary, this thesis contributes a practical and adaptable toolkit for addressing the challenges of transfer learning in a world where the application of machine learning is becoming increasingly widespread. The methodologies and insights we've provided are poised to support the continued growth and diversification of machine learning applications, contributing to the broader use of transfer learning methodologies.

Bibliography

[1] Arif, M., & Mahmood, T. (2015). Cloud computing and its environmental effects. *International Journal of Grid and Distributed Computing*, 8(1), 279-286.

[2] Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2006). Analysis of Representations for Domain Adaptation. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems* (Vol. 19). Retrieved from https://proceedings.neurips.cc/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf

[3] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. (2010). A theory of learning from different domains. *Machine Learning, 79*, 151–175. doi:10.1007/s10994-009-5152-4

[4] Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems, 13*.

[5] Bengio, Y., Bastien, F., Bergeron, A., Boulanger–Lewandowski, N., Breuel, T., Chherawala, Y., ... & Sicard, G. (2011, June). Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (pp. 164-172). JMLR Workshop and Conference Proceedings

[6] Bengio, Y. (2012, June). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning* (pp. 17-36). JMLR Workshop and Conference Proceedings.

[7] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc., 35*, 99–109.

[8] Bellogín, A., & Castells, P. (2010). A performance prediction approach to enhance collaborative filtering performance. In *Advances in Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010*. Proceedings 32 (pp. 382-393). Springer Berlin Heidelberg.

[9] Bingel, J., & Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 2, Short Papers (pp. 164–169). Association for Computational Linguistics.

[10] Boltzmann, L. (1868). Studien uber das gleichgewicht der lebenden kraft. *Wissenschafiliche Abhandlungen*, 1, 49-96.

[11] Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze, 8*, 3-62.

[12] Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, architectures and applications* (pp. 227-236). Berlin, Heidelberg: Springer Berlin Heidelberg.

[13] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.

[14] Burges, C., Ragno, R., & Le, Q. (2006). Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems, 19*.

[15] Burges, C. J. (2010). From RankNet to LambdaRank to LambdaMART: An Overview. *Learning, 11*(23-581), 81.

[16] Caruana, R. (1994). Learning many related tasks at the same time with backpropagation. *Advances in neural information processing systems, 7*.

[17] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794)

[18] Chomsky, N. (1953). Systems of syntactic analysis. *The Journal of Symbolic Logic, 18*(3), 242-256.

[19] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2023). Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research, 24*(240), 1-113.

[20] Chu, C., & Wang, R. (2018). A Survey of Domain Adaptation for Neural Machine Translation. *Proceedings of the 27th International Conference on Computational Linguistics*, 1304–1319. Retrieved from https://aclanthology.org/C18-1111

[21] Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002, August). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 299-306).

[22] Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica, 2*:229-318.

[23] Csiszár, I. (1972). A class of measures of informativity of observation channels. *Periodica Mathematica Hungarica, 2*(1–4), 191–213.

[24] Dai, X., Karimi, S., Hachey, B., & Paris, C. (2019). Using Similarity Measures to Select Pretraining Data for NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers) (pp. 1460–1470). Association for Computational Linguistics.

[25] Dancey, C. P., & Reidy, J. (2007). Statistics without maths for psychology. *Pearson education.*

[26] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), 4171–4186. doi:10.18653/v1/N19-1423

[27] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655). PMLR.

[28] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research, 12*(7).

[29] Elsahar, H., & Galle, M. (2019). To Annotate or Not? Predicting Performance Drop under Domain Shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 2163–2173). Association for Computational Linguistics.

[30] Feydy, J., Séjourne, T., Vialard, F.X., Amari, S.i., Trouve, A., & Peyré, G. (2019). Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (pp. 2681–2690). PMLR.

[31] Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 10-32.

[32] Fukushima, K. (1975). Self-Organizing Multilayered Neural Network. *The Transactions of Electronics and communication Engineers D, 58*(9), 530.

[33] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1), 2096–2030.

[34] Gauss, C., Bertrand, J., & Trotter, H. (1957). *Gauss's Work (1803-1826) on the Theory of Least Squares*. Statistical Techniques Research Group, Section of Mathematical Statistics, Department of Mathematics, Princeton University.

[35] Ian Goodfellow, Yoshua Bengio, & Aaron Courville (2016). Deep Learning. *MIT Press*.

[36] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006). A Kernel Method for the Two-Sample-Problem. In *Advances in Neural Information Processing Systems*. MIT Press.

[37] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

[38] Harris, Z. S. (1954). Distributional structure. *Word, 10*(2-3), 146-162.

[39] He, B., & Ounis, I. (2004, October). Inferring query performance using pre-retrieval predictors. In *International symposium on string processing and information retrieval* (pp. 43-54). Berlin, Heidelberg: Springer Berlin Heidelberg.

[40] Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on, 14*(8), 2.

[41] Hochreiter, Sepp; Schmidhuber, Jürgen (1997-11-01). "Long Short-Term Memory". *Neural Computation. 9* (8): 1735–1780.

[42] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

[43] Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS), 20*(4), 422-446.

[44] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation, 28*(1), 11-21.

[45] Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6(4), 366–422.

[46] Ramesh Kashyap, A., Hazarika, D., Kan, M.-Y., & Zimmermann, R. (2021). Domain Divergences: A Survey and Empirical Analysis. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1830–1849. doi:10.18653/v1/2021.naacl-main.147

[47] Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do Better ImageNet Models Transfer Better?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2661-2671).

[48] S. Kullback, & R. A. Leibler (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79 – 86.

[49] Lange, L., Strotgen, J., Adel, H., & Klakow, D. (2021). To Share or not to Share: Predicting Sets of Sources for Model Transfer Learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 8744–8753). Association for Computational Linguistics.

[50] Lemaréchal, C. (2012). Cauchy and the gradient method. *Doc Math Extra, 251*(254), 10.

[51] Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics, 4*, 151-171.

[52] Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145-151.

[53] Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval, 3*(3), 225-331.

[54] Loshchilov, I., & Hutter, F. (2018). Fixing weight decay regularization in adam.

[55] Lottick, K., Susai, S., Friedler, S. A., & Wilson, J. P. (2019). Energy Usage Reports: Environmental awareness as part of algorithmic accountability. *arXiv preprint arXiv:1911.08354*.

[56] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).

[57] Manning, C.D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. *MIT Press*.

[58] McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation*, 24, 109–165.

[59] McCreadie, R., Buntain, C., & Soboroff, I. (2019). TREC Incident Streams: Finding Actionable Information on Social Media. In *ISCRAM 2019. Proceedings.*

[60] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., ... & Wu, H. (2018, February). Mixed Precision Training. In *International Conference on Learning Representations*.

[61] Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*.

[62] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

[63] Nguyen, C., Hassner, T., Seeger, M., & Archambeau, C. (2020, November). Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning* (pp. 7294-7305). PMLR.

[64] Nivre, J., De Marneffe, M. C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., ... & Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.

[65] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345-1359. doi:10.1109/TKDE.2009.191

[66] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[67] Papoulis, A. (1984) Probability, Random Variables, and Stochastic Processes. McGraw Hill, New York.

[68] Plank, B., & Noord, G. (2011). Effective Measures of Domain Similarity for Parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 1566–1576). Association for Computational Linguistics.

[69] Pfeiffer, J., Ruckle, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., & Gurevych, I. (2020). AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 46–54). Association for Computational Linguistics.

[70] Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., & Gurevych, I. (2021, April). AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 487-503).

[71] Posani, L., Paccoia, A., & Moschettini, M. (2018). The carbon footprint of distributed cloud storage. *arXiv preprint arXiv:1803.06973*.

[72] Poth, C., Pfeiffer, J., Rücklé, A., & Gurevych, I. (2021). What to Pre-Train on? Efficient Intermediate Task Selection. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 10585-10605).

[73] Pugantsov, A., & McCreadie, R. (2023, May). Divergence-Based Domain Transferability for Zero-Shot Classification. In *Findings of the Association for Computational Linguistics: EACL 2023* (pp. 1604-1609).

[74] Purpura, A., Silvello, G., & Susto, G. A. (2022). Learning to rank from relevance judgments distributions. *Journal of the Association for Information Science and Technology, 73*(9), 1236-1252.

[75] Qin, T., & Liu, T. Y. (2013). Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597*.

[76] Paun, I., Moshfeghi, Y., & Ntarmos, N. (2023). White Box: On the Prediction of Collaborative Filtering Recommendation Systems' Performance. *ACM Transactions on Internet Technology, 23*(1), 8:1–8:29.

[77] Phang, J., Févry, T., & Bowman, S. R. (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

[78] Ponomareva, N., & Thelwall, M. (2012, March). Biographies or blenders: Which resource is best for cross-domain sentiment analysis?. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 488-499). Berlin, Heidelberg: Springer Berlin Heidelberg.

[79] Pugantsov, A., & McCreadie, R. (2020). University of Glasgow Terrier Team (uogTr) at the TREC 2020 Incident Streams Track. In *TREC*.

[80] Pugantsov, A., & McCreadie, R. (2021). University of Glasgow Terrier Team (uogTr) at the TREC 2021 Incident Streams Track. In *TREC*.

[81] Puigcerver, J., Riquelme, C., Mustafa, B., Renggli, C., Pinto, A. S., Gelly, S., ... & Houlsby, N. (2020). Scalable Transfer Learning with Expert Models. *arXiv preprint arXiv:2009.13239*.

[82] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[83] Rao, C. R. (1982). Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology*, 21(1), 24–43. doi:10.1016/0040-5809(82)90004-1

[84] Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review*, 97 2, 285–308.

[85] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3982–3992). Association for Computational Linguistics.

[86] Renggli, C., Pinto, A. S., Rimanic, L., Puigcerver, J., Riquelme, C., Zhang, C., & Lučić, M. (2022). Which model to transfer? finding the needle in the growing haystack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9205-9214).

[87] Rényi, A., & Others. (1961). On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1. Berkeley, California, USA.

[88] Rosenstein, M.T., Marx, Z., & Kaelbling, L.P. (2005). To transfer or not to transfer. *Neural Information Processing Systems*.

[89] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, 53-65.

[90] Ruder, S., Ghaffari, P., & Breslin, J. G. (2017). Data Selection Strategies for Multi-Domain Sentiment Analysis. *arXiv preprint arXiv:1702.02426*.

[91] Ruder, S., & Plank, B. (2017). Learning to select data for transfer learning with Bayesian Optimization. In *Proceedings of the 2017 Conference on Empirical Meth-*

ods in Natural Language Processing (pp. 372–382). Association for Computational Linguistics.

[92] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature, 323*(6088), 533-536.

[93] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613-620.

[94] Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.

[95] Sculley, D. (2010). Web-Scale k-Means Clustering. In Proceedings of the 19th International Conference on World Wide Web (pp. 1177–1178). Association for Computing Machinery.

[96] Shannon, C. (1948). A mathematical theory of communication. *The Bell system technical journal, 27*(3), 379–423.

[97] Simpson, E. H. (1949). Measurement of Diversity. *Nature*, 163(4148), 688–688. doi:10.1038/163688a0

[98] So, D., Le, Q., & Liang, C. (2019, May). The evolved transformer. In *International Conference on Machine Learning* (pp. 5877-5886). PMLR.

[99] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation, 28*(1), 11-21.

[100] Spearman, C. (1987). The proof and measurement of association between two things. *The American journal of psychology, 100*(3/4), 441-471.

[101] Steinhaus, H. (1956). Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci, 1*(804), 801.

[102] Strubell, E., Ganesh, A., & McCallum, A. (2019, July). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645-3650).

[103] Baochen Sun, Jiashi Feng, & Kate Saenko (2016). Correlation Alignment for Unsupervised Domain Adaptation. CoRR, abs/1612.01939.

[104] Tsvetkov, Y., Faruqui, M., Ling, W., MacWhinney, B., & Dyer, C. (2016). Learning the Curriculum with Bayesian Optimization for Task-Specific Word Representation Learning. In *Proceedings of the 54th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers) (pp. 130–139). Association for Computational Linguistics.

[105] Uspensky, J. V. (1937). Introduction to mathematical probability. *McGraw-Hill*.

[106] Van Asch, V., & Daelemans, W. (2010). Using Domain Similarity for Performance Estimation. *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 31–36. Retrieved from https://aclanthology.org/W10-2605

[107] Wong, A. K. C., & You, M. (1985). Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5), 599–609.

[108] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

[109] Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., & Iyyer, M. (2020). Exploring and Predicting Transferability across NLP Tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (EMNLP) (pp. 7882–7926). Association for Computational Linguistics.

[110] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (pp. 353–355). Association for Computational Linguistics.

[111] Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM, 9*(1), 36-45.

[112] Wu, F., & Huang, Y. (2016). Sentiment Domain Adaptation with Multiple Sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 301–310). Association for Computational Linguistics.

[113] Xia, M., Anastasopoulos, A., Xu, R., Yang, Y., & Neubig, G. (2020, July). Predicting Performance for Natural Language Processing Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8625-8646).

[114] You, K., Liu, Y., Wang, J., & Long, M. (2021, July). Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning* (pp. 12133-12143). PMLR.

[115] Zamir, Amir R., Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. "Taskonomy: Disentangling task transfer learning." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712-3722. 2018.

[116] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems, 28.*

[117] Werner Zellinger, Bernhard A. Moser, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, & Susanne Saminger-Platz (2019). Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, 483, 174-191.

Appendix A

Parameters

A.1 Aggregation Hyperparameters

Table A.1: Hyperparameter space used in our Bayesian search for aggregating representations for instance-to-instance ($INST - INST$), centroid-to-domain ($CTD - DOM$), and centroid-to-centroid ($CTD - CTD$) comparisons in Chapters 6-7.

Hyperparameter	**Value Ranges**
K	$\{25n : n \in \mathbb{N}, 1 \leq n \leq 5\}$
$K_{\mathcal{D}_I}$	$\{500n : n \in \mathbb{N}, 2 \leq n \leq 10\}$
$K_{\mathcal{D}_T}$	$\{50n : n \in \mathbb{N}, 2 \leq n \leq 20\}$

A.2 Task Selection Hyperparameters

Table A.2: Hyperparameter space used in our Bayesian search for intermediate task selection models (regression and ranking) in Chapter 8.

Hyperparameter	**Value Ranges**
`learning_rate`	0.01, 0.05, 0.1, 0.2
`max_depth`	3, 6, 9, 12
`min_child_weight`	1, 3, 5, 7
`gamma`	0, 0.5, 1, 2
`subsample`	0.5, 0.7, 0.9, 1
`colsample_bytree`	0.5, 0.7, 0.9, 1
`colsample_bylevel`	0.5, 0.7, 0.9, 1
`lambda`	0, 0.5, 1, 2
`alpha`	0, 0.5, 1, 2
Regression	
`n_estimators`	0, 0.5, 1, 2
Ranking	
`num_boost_rounds`	0, 0.5, 1, 2