METHODS FOR LEVERAGING AUXILIARY SIGNALS FOR LOW-RESOURCE NLP

By

XIN DONG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Gerard de Melo

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January 2023

ABSTRACT OF THE DISSERTATION

Methods for Leveraging Auxiliary Signals

for Low-Resource NLP

by XIN DONG

Dissertation Director: Gerard de Melo

There is a growing need for NLP systems that support low-resource settings, for which task-specific training data may be lacking, while domain-specific corpora is too scarce to build a reliable system. In the past decade, the co-occurrence-based training objectives of methods such as word2vec are first able to offer word-level semantic information for specific domains. Recently, pretrained language model architectures such as BERT have been shown capable of learning monolingual or multilingual representations with self-supervised objectives under a shared vocabulary, simply by combining the input from single or multiple languages. Such representations greatly facilitate low-resource language applications. Still, the success of such cross-domain transfer hinges on how close the involved domains are, with substantial drops observed for some more distant domain pairs, such as English to Korean, Wikipedia text to social media comments. To address this, domain-specific unlabeled corpora is available to serve as the auxiliary signals to enhance low-resource NLP systems. In this dissertation, we present a series of methods for leveraging auxiliary signals. In particular, cross-lingual sentiment embeddings with transfer learning are proposed to improve sentiment analysis. For cross-lingual text classification, we present a self-learning framework to take advantage of unlabeled data. Furthermore, a framework upon data augmentation with adversarial training for cross-lingual NLI is proposed for the

low-resource problem from the target domain. Finally, we present two effective methods on injecting extra information with auxiliary signals from multiple sources for temporal event reasoning and rating estimation in recommendation system. Extensive experimental results demonstrate the effectiveness of the proposed methods in achieving better performance across a variety of NLP tasks.

ACKNOWLEDGMENTS

First of all, I would like to express my truthful gratitude to my advisor Prof. Gerard de Melo for his guidance during my whole PhD study. I still remembered the day I took my co-authored paper to his office for self-introduction on the research as a rookie. His kind and considerate attitude relieved my nervousness and his detailed enlightenment started this journey. Over the past 5 years, he consistently taught me how to be a good researcher from discovering, formulating a research problem to finishing a solid paper. Without his guidance, it is unlikely to complete my Ph.D study and this dissertation.

I am also thankful to the Department of Computer Science, Prof. Gerard de Melo, and JPMorgan Chase for funding my research. I thank Adobe, NEC Lab Amercica, Dataminr, Pinterest for providing me the opportunities to carry out my research and experience industry work. I want to thank Prof. Yongfeng Zhang, Prof. Karl Stratos and Dr. Handong Zhao for serving on my defense committee and providing me with insightful feedback on my thesis work.

It was my honor to work with talented collaborators, including Sen Yang who leaded me to the research career, Zuohui Fu, Yaxin Zhu, Dongkuan Xu, Jingchao Ni, Tanay Kumar Saha, Ke Zhang, Joel Tetreault. I thank them for fruitful collaborations or dedicated mentorship on various projects.

Finally, I would like to thank my parents for their unconditional support. I also want to express my gratitude to my wife Ruiqi Wang, who accompanys me through the journey. Her support and encouragement enable me to overcome difficulties. She is like a star in the dark night that never makes me feel lost in life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Based on massive amounts of data, recent pretrained contextual representation models have made significant strides in advancing a number of different NLP tasks, such as on text classification, reading comprehension, etc. However, for some NLP tasks in low-resources settings, relevant training data may be lacking, while state-of-the-art deep learning methods are known to be data-hungry. For example, given the rapid progress of globalization, multinationals such as Google, Meta, Amazon are increasingly exploring new opportunities in different markets across the globe. Because of this, they are increasingly also expected to be able to provide multilingual support for application interfaces, product manuals, search services, name-entity recognition, etc. For Instagram specifically, one important task is to detect bad comments to maintain a good atmosphere in a global comment zone. The key target here is to monitor comments and demote some potentially toxic comments. For instance, when there is an offensive comment about the creator, demotion should be automatically done. Unfortunately, the cost of acquiring new custom-built resources for each combination of language and domain is very high, as it typically requires human annotation. Available resources for domain-specific tasks are often imbalanced between different languages. The scarcity of non-English annotated corpora may preclude our ability to train language-specific machine learning models. In contrast, English-language annotations are often readily available to train deep models. Although translation can be an option, human translation is very costly and for many language pairs, any available domain-specific parallel corpora are too small to train high-quality machine translation systems.

To overcome this, cross-domain systems can be trained on a source domain and subsequently also be applied to other domains despite a complete lack of labelled training

data for those target domains. Specifically, in the past, cross-lingual systems typically drew on translation dictionaries, lexical knowledge graphs, or parallel corpora, to build a basic cross-lingual text classification model that exploits connections between words and phrases across different languages [1]. Recently, pretrained language model such as BERT [2] have been shown capable of learning joint multilingual representations with self-supervised objectives under a shared vocabulary, simply by combining the input from multiple languages [3, 2, 4, 5]. With multilingual models [3, 2], substantial progress has been achieved in cross-lingual training, even for distant language pairs such as English and Korean. Another available solution is auxiliary learning, which is also widely used to integrate extra information into the primary task. The role of the auxiliary tasks is to improve the performance and generalizability of this primary task.

In general, for low-resource NLP application, from not only multilingual support, but also temporal event detection, recommendation system, etc, auxiliary signals obtained from different data resources can always be considered to further improve the performance. In this dissertation, we focus on presenting different available methods for leveraging auxiliary signals for low-resource NLP.

1.1 Contributions

The main contributions of this dissertation are summarized in the following.

Cross-Lingual Sentiment Propagation and Transfer Learning. Deep convolutional neural networks excel at sentiment polarity classification, but tend to require substantial amounts of training data, which moreover differs quite significantly between domains. In this work, we present an approach to project sentiment information across languages. We propose encoding this information in embedding vectors that capture sentiment properties along multiple dimensions and allow the model to adapt to different domains and circumstances. This is different from previous work on cross-lingual projection, which has considered generic sentiment polarity lexicons. Different words, however, may have strikingly different

connotations in different contexts. For instance, *hot* is generally positive when referring to music, but tends to be negative when referring to the temperature in a hotel room. We incorporate the induced embeddings into the model via a dedicated memory-based component, and show that our approach can lead to consistent gains across different languages on diverse datasets from different domains.

Self-Learning in Cross-Lingual Text Classification. Recent pretrained multilingual representation models make it possible to exploit labeled data from one language to train a cross-lingual text classification model that can then be applied to a completely different language. However, there may still be subtle differences between languages that are neglected when doing so. To address this, we present a semi-supervised adversarial training process that minimizes the maximal loss for label-preserving input perturbations. Our model begins by learning just from available source language samples, drawing on a multilingual encoder with the added adversarial perturbation. Without loss of generality, in the following, we assume English to be the source language. After training on English, subsequently, we use the same model to make predictions on unlabeled non-English samples and a part of those samples with high confidence prediction scores are repurposed to serve as labeled examples for a next iteration of adversarial training until the model converges.

The adversarial perturbation added during self-learning improves robustness and generalization by regularizing our model. At the same time, because adversarial training makes tiny perturbations that barely affect the prediction result, the perturbations on words during self-learning can be viewed as inducing a form of code-switching, which replaces some original source language words with potential nearby non-English word representations.

Based on this combination of adversarial training and semi-supervised self-learning techniques, the model evolves to become more robust with regard to differences between languages. We demonstrate the superiority of our framework on Multilingual Document Classification (MLDoc) [6] in comparison with state-of-the-art baselines. Our study then proceeds to show that our method outperforms other methods on cross-lingual dialogue intent

classification from English to Spanish and Thai [7]. This shows that our semi-supervised adversarial framework is more effective than previous approaches at cross-lingual transfer for domain-specific tasks, based on a mix of labeled and unlabeled data via adversarial training on multilingual representations.

Data Augmentation with Adversarial Training for Cross-Lingual NLI. In this part, we focus on cross-lingual sentence-pair classification, an important class of problems that involve determining the kind of relationship between two input texts. This encompasses numerous tasks in NLP, e.g., natural language inference (NLI) and duplicate/paraphrase detection. Such tasks typically require expensive human-labeled training data, and the model needs to carefully discern pertinent connections between two inputs, so cross-lingual models are often fairly brittle. Therefore, we propose a novel data augmentation strategy for better cross-lingual natural language inference by enriching the data to reflect more diversity in a semantically faithful way. To this end, we propose two methods of training a generative model to induce synthesized examples, and then leverage the resulting data using an adversarial training regimen for more robustness. In a series of detailed experiments, we show that this fruitful combination leads to substantial gains in cross-lingual inference.

Multi-Source Auxiliary Learning in Temporal Event Reasoning. Temporal event reasoning is vital in modern information-driven applications operating on news articles, social media, financial reports, etc. Recent works train deep neural nets to infer temporal events and relations from text. We improve upon the state-of-the-art by proposing an approach that injects additional temporal knowledge into the pre-trained model from two sources: (*i*) part-of-speech tagging and (*ii*) question constraints. Auxiliary learning objectives allow us to feed this temporal information into the training process. Our experiments show that these types of multi-source auxiliary learning objectives lead to better temporal reasoning. Our model improves over the state-of-the-art model on the TORQUE question answering and the MATRES relation extraction benchmark.

Auxiliary Asymmetrical Hierarchical Review-Based Networks for Rating Estimation.

Recommender systems have been able to emit substantially improved recommendations by leveraging user-provided reviews as the auxiliary signals. Existing methods typically merge all reviews of a given user or item into a long document, and then process user and item documents in the same manner. In practice, however, these two sets of reviews are notably different: users' reviews reflect a variety of items that they have bought and are hence very heterogeneous in their topics, while an item's reviews pertain only to that single item and are thus topically homogeneous. In this work, we develop a novel neural network model that properly accounts for this important difference by means of asymmetric attentive modules. The user module learns to attend to only those signals that are relevant with respect to the target item, whereas the item module learns to extract the most salient contents with regard to properties of the item. Our multi-hierarchical paradigm accounts for the fact that neither are all reviews equally useful, nor are all sentences within each review equally pertinent. Extensive experimental results on a variety of real datasets demonstrate the effectiveness of our method.

1.2 Dissertation Outline

The remaining chapters of this dissertation are organized as follows. Chapter 2 presents an overview of the relevant background. Chapter 3 presents a cross-lingual propagation algorithm that yields sentiment embedding vectors as the auxiliary signals for transfer learning in numerous languages . Chapter 4 explores a self-learning framework in cross-lingual text classification. Chapter 5 contains data augmentation with adversarial training for cross-lingual NLI. Chapter 6 introduces a novel multi-source auxiliary learning in temporal event reasoning. Chapter 7 presents a novel neural recommendation model including user's and item's review sets that properly accounts for this important difference by means of asymmetric attentive modules. Concluding remarks and a discussion on future research directions are presented in Chapter 8.

CHAPTER 2
BACKGROUND

This chapter consists of a brief overview on sentiment embedding, cross-lingual systems, temporal event reasoning and review-based recommendation system for understanding the backgrounds about our problems.

2.1 Sentiment Embedding

Word embedding methods such as word2vec [8] are now ubiquitously used across a wide range of tasks in the broad area of text mining and natural language processing, including in models for sentiment analysis [9, 10, 11]. Cross-lingual distributed representations have been studied as well. These are typically produced either by aligning multiple monolingual word embedding models using techniques such as linear projections [12] or CCA [13], by jointly training in multiple languages via parallel corpora [14, 15], or by exploiting multilingual semantic resources [16, 17]. However, the co-occurrence-based training objectives of methods such as word2vec do not consider sentiment specifically. Some work, in contrast, focuses on representations that capture sentiment-specific cues rather than generic word semantics.

Sentiment Lexicons. Despite their inherent limitations, lexicon-driven sentiment analysis methods remain widespread. One of their advantages is that they may be better-suited at performing robustly across different domains compared to supervised approaches, which may pick up dataset-specific correlations. The latter, for instance, may learn that mentions of the word *novel* in movie reviews often correlate with lower review scores due to movies not living up to the expectations of fans of the novel. We thus consider English sentiment lexicons as a simple baseline form of English vector representations. Specifically, we rely on a recent sentiment lexicon called VADER [18], and view the polarity scores that it assigns

to words as components of simple 1-dimensional word vectors.

Domain-Specific Lexicon Induction. Generic sentiment lexicons do not account for the domain-specific nature of word polarity scores. A word that has positive connotations in one domain may have negative connotations in another domain. We hence consider the SocialSent Reddit community-specific data mined by the Stanford NLP group [19]. Their study produced separate domain-specific scores for each of 250 different subcommunities of the Reddit social media forum site. Although this data is biased by its source and by their semi-automatic induction process, we consider it a valuable resource. Taken together, the 250 different lexicons can be used to induce 250-dimensional vector embeddings that reflect the distribution of a word's sentiment polarity across a large range of domains.

2.2 Cross-lingual Systems

Owing to notable advances in deep learning and representation learning, important progress has been achieved on text classification, reading comprehension, and other NLP tasks. Recently, pretrained language representations with self-supervised objectives [20, 2, 21] have further pushed forward the state-of-the-art on many English tasks. While these sorts of deep models can be trained on different languages, deep models typically require substantial amounts of labeled data for the specific domain of data.

Unfortunately, the cost of acquiring new custom-built resources for each combination of language and domain is very high, as it typically requires human annotation. Available resources for domain-specific tasks are often imbalanced between different languages. The scarcity of non-English annotated corpora may preclude our ability to train language-specific machine learning models. In contrast, English-language annotations are often readily available to train deep models. Although translation can be an option, human translation is very costly and for many language pairs, any available domain-specific parallel corpora are too small to train high-quality machine translation systems.

Cross-lingual systems rely on training data from one language to train a model that

can be applied to other languages [1], alleviating the training bottleneck issues for low-resource languages. This is facilitated by recent advances in learning joint multilingual representations [2, 4, 5].

2.3 Data Augmentation

Data augmentation is a promising technique, especially when dealing with scarce data, imbalanced data, or semi-supervised learning problems. Back-translation [22] has been considered as a technique to obtain alternative examples preserving the original semantics, by translating an existing example in language L_A into another language L_B and then translating it back into L_A to obtain an augmented example. Yu *et al.* [23] and Xie *et al.* [24] applied it to question answering and semi-supervised monolingual training scenarios. However, this requires high-quality translation engines that often do not exist in the settings in which one wishes to apply cross-lingual systems.

Wei and Zou [25] instead combined synonym replacement, random insertion, random swapping, and random deletion in a method named EDA. Since insertion and deletion may affect the semantics of the utterance, some studies opt to control the selection of words to be replaced with indicators such as TF-IDF scores [24]. Fadaee *et al.* [26] use contextualized word embeddings to replace the target word. Kobayashi [27] proposed a bi-directional language-model-based augmentation method, and Wu *et al.* [28] further improved its results by switching to BERT. Another major category is text generation based augmentation. Anaby-Tavor *et al.* [29] proposed a language model based data augmentation method, shown to improve classifier performance on a variety of English datasets. It relies on GPT-2 [21] to generate a single new sequence in each instance.

2.4 Temporal Event Reasoning

Temporal event reasoning is a crucial yet under-explored aspect of interpreting text in modern information systems, enabling people to infer the timeline of narrated events. Past work has

often cast this as a Relation Extraction task [30, 31, 32] that involves predicting temporal relationships between two events mentioned in a given piece of text, such as BEFORE or AFTER. Another recently proposed task is that of reading comprehension about temporal relations [33]. Given an input text, the system answers temporal questions pertaining to some event. Compared with the aforementioned temporal relationship prediction task, the advantage of such a Question Answering (QA) problem formulation is that questions can encode a richer, more diverse range of complex temporal relationships and phenomena, such as overlap, uncertainty, negation, hypotheticals, and repetition, to name a few. For instance, we may ask a challenging question incorporating negation such as "What has not happened after investigators made good progress?"

2.5 Review-Based Recommendation System

The rapid shift from traditional retail and services to online transactions has brought forth a large volume of review data in areas such as e-commerce, dining, tourism, among many others. While such reviews are routinely consulted directly by consumers and affect their decision making, recent work has shown that they can also be exploited by intelligent algorithms. The detailed semantic cues that they harbor not only reveal different aspects (*e.g.*, quality, material, color, *etc.*) of an item, but also reflect the sentiment of users towards these aspects. Such fine-grained auxiliary signals are extremely valuable to a recommender system and significantly complement the sparse rating and click-through data, based on which many traditional collaborative filtering methods [34] have been developed. Thus, there has been a series of studies seeking to harness the potential of reviews in improving the recommendation quality [35, 36, 37, 38].

These studies have shown that leveraging reviews can indeed boost the recommendation effectiveness quite remarkably. Typically, they associate users with the respective sets of reviews they have written, while associating each item with the set of all reviews that have been written for it. To predict the rating for an unseen user–item pair, in a first step, the

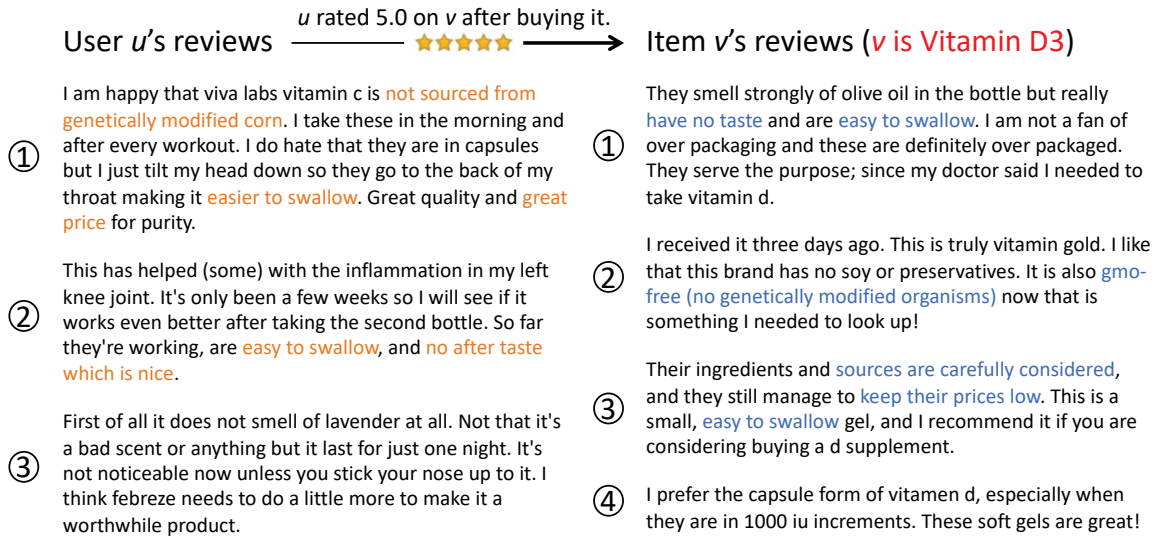

Figure 2.1: An example of reviews by a user and for an item. User u rated a 5.0 score on item v after purchasing it.

embeddings of that user and item are inferred from the respective sets of reviews via a neural network. Then, the two embeddings are matched to predict a numeric rating between them. For example, DeepCoNN [35] relies on convolutional neural networks to learn user (item) embeddings, and on a factorization machine [39] to predict ratings. D-ATT [37] uses dual-attention based networks to learn embeddings, and a simple dot product to predict ratings.

Despite the encouraging progress, existing methods all regard the set of reviews by a user and the set of reviews for an item as the same type of documents, and invoke the same model (or even a shared model) to process them in parallel. In reality, however, the set of reviews for a user is fundamentally different from the set of reviews for an item. In particular, reviews for users correspond to a diverse set of items that they have rated, resulting in notably *heterogeneous* textual contents with a variety of topics for different items. In contrast, each item's reviews are only about itself, and the contents are thus *homogeneous* in the sense that the topic is limited to a single narrow domain. For example, Figure 2.1 shows several reviews from Amazon's health domain. User u's historical reviews describe three items, Vitamin C, anti-inflammatory medication, and an air freshener, while all reviews for item v

are about itself, *i.e.*, Vitamin D3.

This profound difference necessitates distinct forms of attention to be paid on user reviews as opposed to item reviews, when deciding whether to recommend an item v to a user u. To predict u's preference of v, it is important to extract from u's reviews those aspects that pertain most to v, e.g., comments on items that are similar to v. In contrast, from v's reviews, we wish to account for the sentiment of other users with regard to relevant aspects of v. If u pays special attention to certain aspects of items similar to v, while other users wrote highly about v with regard to these particular aspects, then it is much more likely that v will be of interest to u. For example, in Figure 2.1, reviews 1 and 2 of u are about non-prescription medicines that are similar to v. In reviews 1 and 2, u mentioned aspects such as "not sourced from genetically modified corn", "easier to swallow", "great price", and "no after taste", indicating that u considers the source and price and prefers easily swallowed products without after-taste. Meanwhile, reviews 1-3 of v mention that v "have no taste", is "easy to swallow", "gmo-free", and "prices low", which are opinions expressed by others that match u's preferences. Thus, v is likely to be of interest to u, and u indeed marked a 5.0 score on v after purchasing it.

Another vital challenge is how to reliably represent each review. Importantly, sentences are not equally useful within each review. For example, in Figure 2.1, the second sentence in u's review 1, "*I take these in the morning and after every workout.*" conveys little regarding u's concerns for Vitamin C, and thus is less pertinent than other sentences in the same review. Since including irrelevant sentences can introduce noise and may harm the final embedding quality, it is crucial to aggregate only useful sentences to represent each review.

CHAPTER 3
CROSS-LINGUAL SENTIMENT PROPAGATION AND TRANSFER LEARNING

3.1 Overview

As more and more users come online across the globe, increasing numbers of people are voicing their opinion in social media, blogs, review sites, and other online fora. Given such valuable data, modern deep learning-based sentiment analysis methods excel at determining the sentiment polarity of what is being said about companies, products, etc. [40]. Unfortunately, such deep methods require substantial amounts of training data, because multiple levels of computation, each with additional weights and parameters, need to be learned, typically via end-to-end training. This is a significant problem for many of the world's languages, for which resources may be too costly to obtain and training data is scarce, especially when one considers that new training data is needed for each domain and genre. A model trained on movie reviews, for instance, will fare very poorly on the task of assessing digital camera reviews, let alone social media postings such as tweets.

For many languages and domains, there is a paucity of available data and resources. In some cases, it may be challenging to obtain sufficient in-domain training data, both because there may be less data available online and because it may be somewhat harder to find annotators. Hence, a question that arises is whether one can assist deep networks by incorporating external cues. We conjecture that vector representations are a suitable means of injecting sentiment-related signals into neural models, as a sort of external prior. Generic word vectors as produced by word2vec [8] are widely used to feed generic semantic information into a model. Preinitialization with such vectors often leads to noticeable gains compared to randomly initialized embedding matrices[10]. Therefore, we consider the question of whether further gains can be achieved by relying on cross-lingual induction

to obtain more targeted signals pertaining to a word's sentiment rather than to its general semantics.

In this chapter, we present a cross-lingual propagation algorithm to overcome these challenges and enable improved deep sentiment analysis across a range of languages and domains. Our approach relies on word vectors that are cross-lingually projected from a source language such as English to any number of target languages. We present an approach to project sentiment information across languages. We propose encoding this information in embedding vectors that capture sentiment properties along multiple dimensions and allow the model to adapt to different domains and circumstances. This is different from previous work on cross-lingual projection, which has considered generic sentiment polarity lexicons. Different words, however, may have strikingly different connotations in different contexts. For instance, *hot* is generally positive when referring to music, but tends to be negative when referring to the temperature in a hotel room. After obtaining sentiment word embedding, an intuitive solution would be to concatenate regular embeddings, which provide semantic relatedness cues, with sentiment polarity cues that are captured in additional dimensions. We instead propose a bespoke convolutional neural network architecture with a separate memory module dedicated to the sentiment embeddings. Our empirical study shows that the sentiment embeddings can lead to consistent gains across different datasets in a diverse set of domains and languages if a suitable neural network architecture is used.

3.2 Related Work

Cross-Lingual Sentiment Analysis. The majority of research on sentiment analysis has focused on the English language. One way of supporting further languages is to use machine translation, as has been investigated for subjectivity [41] and sentiment polarity [42]. However, this may be overly computationally intensive when analyzing the vast quantities of data posted online. Moreover, Duh *et al.* [43] argued that even perfect machine translation incurs a degradation in the result quality for sentiment analysis, while showing

that regular adaptation methods do not work well in this setting. Haas and Versley [44] provided empirical support in line with these claims. Another option, proposed by Vilares *et al.* [45], is to forgo supervision from training data, instead relying on rules applied to syntactic dependencies. Wan [46] presented a bilingual co-training approach that jointly trains a system on two languages, considering each language an independent view.

An alternative strategy is to use cross-lingual projection, which involves transferring annotations from a source language resource to some target language by exploiting translational equivalence [47, 48] or parallel corpora [49]. There are several English-language sentiment lexicons, many of which have been compiled manually [50] or via crowdsourcing [51]. While these are costly to produce, one can subsequently use cross-lingual projection techniques to effectively translate such lexicons to new languages. Mihalcea *et al.* [52] proposed an approach to achieve this for subjectivity lexicons.

Boyd-Graber and Resnik [53] proposed a cross-lingual probabilistic generative model for sentiment analysis. Balamurali *et al.* [54] use cross-lingual projection by means of word sense disambiguation, but the approach hinges on the existence of multilingual wordnets that map words in different languages to a shared interlingual representation. In terms of broad multilingual support, the most relevant previous work is that of Chen and Skiena [55], which used joint cross-lingual propagation to create sentiment lexicons for dozens of languages. We compare against these in our experiments.

An important shortcoming of sentiment lexicons is that they neglect the domain-specific nature of word sentiment polarities. For instance, a word such as *scary* tends to be negative, but may also correlate with positive movie review scores. Our work, in contrast, focuses on multi-dimensional word representations for deep neural networks.

Mining Sentiment Information. There are various monolingual methods to mine sentiment information. For instance, one can collect reviews that come with associated ratings, and use supervised learning to learn feature weights [56]. One can also apply distant supervision exploiting the presence of emoticons or hashtags in online social media [57]. In our work,

we as well start off with such approaches to obtain initial English data, and then rely on a cross-lingual induction procedure to transfer the acquired representations to new language.

Neural Architectures. In terms of architectures, deep recursive neural networks [9] were soon outperformed by deep convolutional and recurrent neural networks [10, 58]. Recent work has investigated more involved models, with ingredients such as Tree-LSTMs [59, 60], hierarchical attention [61], user and product attention [62], aspect-specific modeling [40], and part of speech-specific transition functions [63]. Large ensemble models also tend to outperform individually trained sentiment analysis models [60]. The goal of our study is not necessarily to devise the most sophisticated state-of-the-art neural architecture, but to demonstrate how external sentiment cues can be incorporated such architectures. Our initial explorations relied on a simple dual-channel convolutional neural network [64]. The present work proposes a more sophisticated approach, drawing on ideas from attention mechanisms in machine translation [65] as well as from memory networks [66] and iterative attention [67], which have proven useful for tasks such as question answering. We incorporate these ideas into a separate memory module that operates alongside the regular convolutional module.

3.3 Sentiment Embedding Computation

Our goal is to incorporate external cues into a deep neural network such that the network is able to generalize better even when training data is scarce. While in computer vision, weights pre-trained on ImageNet are often used for transfer learning, the most popular way to incorporate external information into deep neural networks for text is to draw on word embeddings trained on vast amounts of word context information [68, 69, 70]. Indeed, the semantic relatedness signals provided by such representations often lead to slightly improved results in polarity classification tasks [9, 10, 11].

However, the co-occurrence-based objectives of word2vec and GloVe do not consider sentiment specifically. We thus seek to examine how complementary sentiment-specific

information from an external source can give rise to further gains.

Transfer Learning. To this end, our goal is to induce sentiment embeddings that capture sentiment polarity signals in multiple domains and hence may be useful across a range of different sentiment analysis tasks. The multi-domain nature of these distinguish them from the kinds of generic polarity scores captured in sentiment polarity lexicons. We achieve this via transfer learning from trained models, benefiting from supervision on a series of sentiment polarity tasks from different domains. Given a training collection consisting of n binary classification tasks (e.g., with documents in n different domains), we learn n corresponding polarity prediction models. From these, we then extract token-level scores that are tied to specific prediction outcomes. Specifically, we train n linear models $f_i(\mathbf{x}) = \mathbf{w}_i^{|}\mathbf{x} + b_i$ for tasks $i = 1, \ldots, n$. Then, each vocabulary word index j is assigned a new n-dimensional word vector $\mathbf{x_j} = (w_{1,j}, \quad , w_{n,j})$ that incorporates the linear coefficients for that word across the different linear models.

A minor challenge is that naïvely using bag-of-word features can lead to counter-intuitive weights. If a word such as "*pleased*" in one domain mainly occurs after the word "*not*", while the reviews in another domain primarily used "*pleased*" in its unnegated form, then "*pleased*" would be assessed as possessing opposite polarities in different domains. To avoid this, we assume that features are preprocessed to better reflect whether words occur in a negated context. In our experiments, we simply treat occurrences of "*not* hword*i*" as a single feature "*not_*hword*i*". Of course, one can replace this heuristic with much more sophisticated techniques that fully account for the scope of a wider range of negation constructions.

Graph-Based Extension. Most sentiment-related resources are available for the English language. To produce vectors for other languages in our experiments, we rely on cross-lingual projection via graph-based propagation [16, 17, 64]. At this point, we have a set of initial sentiment embedding vectors $\tilde{\mathbf{v}}_x$ *2* R^n for words x *2* V_0. We assume that we have a lexical knowledge graph $G_{\mathrm{L}} = (V, A_{\mathrm{L}})$ with a node set consisting of an extended multilingual vocabulary $V \quad V_0$ and a set of weighted directed arcs $A_{\mathrm{L}} = \mathsf{f}(x_1, x_1^{\emptyset}, w_1), \ldots, (x_m, x_m^{\emptyset}, w_m)\mathsf{g}$.

Each such arc reflects a weighted semantic connection between two vocabulary items $x, x' \in V$, where vocabulary items are words labeled with their respective language. Typically, many of the arcs in the G_{L} would reflect translational equivalence, but in our experiments, we also include monolingual links between semantically related words. Given this data, we aim to minimize

$$\sum_{x \in V} \mathbf{v}_x \left[\frac{1}{\sum_{(x,x',w) \in A_{\mathrm{L}}} w} \sum_{(x,x',w) \in A_{\mathrm{L}}} w \mathbf{v}_{x'} \right] + C \sum_{x \in V_0} \| \mathbf{v}_x - \tilde{\mathbf{v}}_x \|_2 \tag{3.1}$$

The first component of this objective seeks to ensure that sentiment embeddings of words accord with those of their connected words, in terms of the dot product. The second part ensures that the deviation from any available initial word vectors $\tilde{\mathbf{v}}_x$ is minimal (for some very high constant C). For optimization, we preinitialize $\mathbf{v}_x = \tilde{\mathbf{v}}_x$ for all $x \in V_0$, and then rely on stochastic gradient descent steps.

3.4 Dual-Module Memory based CNNs

To feed this sentiment information into our architecture, we propose a Dual-Module Memory based Convolutional Neural Network (DM-MCNN) approach, which incorporates a dedicated memory module to process the sentiment embeddings, as illustrated in Figure 3.1. While the module with regular word embeddings enables the model to learn salient patterns and harness the nearest neighbour and linear substructure properties of word embeddings, we conjecture that a separate sentiment memory module allows for better exploiting the information brought to the table by the sentiment embeddings.

Convolutional Module Inputs and Filters. The Convolutional Module input of the DM-MCNN is a sentence matrix $\mathbf{S} \in \mathbb{R}^{s \times d}$, the rows of which represent the words of the input sentence after tokenization. In the case of $\mathbf{S}$, i.e., in the regular module, each word is

Figure 3.1: (a) Dual-Module Memory based Convolutional Neural Network architecture. (b) Single layer in Memory Module

represented by its conventional word vector representation. Here, s refers to the length of a sentence, and d represents the dimensionality of the regular word vectors.

We perform convolutional operations on these matrices via linear filters. Given rows representing discrete words, we rely on weight matrices $\mathbf{W} \in \mathsf{R}^{h \times d}$ with region size h. We use the notation $\mathbf{S_{i:j}}$ to denote the sub-matrix of $\mathbf{S}$ from row i to row j. Supposing that the weight matrix has a filter width of h, a wide convolution [71] is induced such that out-of-range submatrix values $S_{i,j}$ with $i < 1$ or $i > s$ are taken to be zero. Thus, applying the filter on sub-matrices of $\mathbf{S}$ yields the output sequence $\mathbf{o} \in \mathsf{R}^{s+h-1}$ as

$$o_i = \mathbf{W} \quad \mathbf{S_{i:i+h-1}}, \tag{3.2}$$

where the operator provides the sum of an element-wise multiplication. Wide convolutions ensure that filters can cover words at the margins of the normal weight matrix.

Next, the c_i in feature maps $\mathbf{c} \in \mathsf{R}^{s+h-1}$ are computed as: $c_i = f(o_i + b)$, where $i = 1, \ldots, s + h \quad 1$, the parameter $b \in \mathsf{R}$ is a bias term, and f is an activation function.

Multiple Layers in Memory Module. The memory module obtains as input a sequence of sentiment embedding vectors for the input, and attempts to draw conclusions about the overall sentiment polarity of the entire input sequence. Given a set of sentence words $S = \{w_1, w_2, w_3, \ldots, w_n\}$, each word is mapped to its sentiment embedding vector of dimension d_s and we denote this set of vectors as V_s. The preliminary sentiment level $\mathbf{v}_\mathrm{p}$ is also a vector of dimensionality d_s. We take the mean of all sentiment vectors $\mathbf{v}_i$ for words $w_i \in S$ to initialize $\mathbf{v}_\mathrm{p}$. Next, we compute a vector $\mathbf{s}$ of similarities s_i between $\mathbf{v}_\mathrm{p}$ and each sentiment word vector $\mathbf{v}_i$, by taking the inner product, followed by ℓ_2-normalization and a softmax:

$$s_i = \frac{\exp \frac{\mathbf{v}_\mathrm{p}^\intercal \mathbf{v}_i}{\|\mathbf{v}_\mathrm{p}^\intercal \mathbf{v}_i\|_2}}{\sum_i \exp \frac{\mathbf{v}_\mathrm{p}^\intercal \mathbf{v}_i}{\|\mathbf{v}_\mathrm{p}^\intercal \mathbf{v}_i\|_2}} \tag{3.3}$$

As the sentiment embeddings used in this chapter are generated from a linear model, the degree of correspondence between $\mathbf{v}_\mathrm{p}$ and $\mathbf{v}_i$ can adequately be assessed by the inner product. The resulting vector of scores $\mathbf{s}$ can be regarded as yielding sentiment weights for each word in the sentence. We apply ℓ_2-normalization to ensure a more balanced weight distribution. The output sentiment level vector $\mathbf{v}_\mathrm{o}$ is then a sum over the sentiment inputs $\mathbf{v}_i$ weighted by the ℓ_2-normalized vector of similarities:

$$\mathbf{v}_\mathrm{o} = \sum_i \frac{s_i}{\|\mathbf{s}\|_2} \mathbf{v}_i \tag{3.4}$$

This processing can be repeated in multiple passes, akin to how end-to-end memory networks for question answering often perform multiple hops [72]. While in the first iteration, $\mathbf{v}_\mathrm{p}$ was set to the mean sentiment vector, subsequent passes may allow us to iteratively refine this vector. Assuming that $\mathbf{v}_\mathrm{o}^k$ has been produced by the k-th pass, then the subsequent level $\mathbf{v}_\mathrm{p}^{k+1}$ in the next pass is:

$$\mathbf{v}_\mathrm{p}^{k+1} = \mathbf{v}_\mathrm{o}^k + \mathbf{v}_\mathrm{p}^k \tag{3.5}$$

The intuition here is that multiple passes can enable the model to adaptively retrieve iterative sentiment level statistics beyond the initial average sentiment information.

Merging Layer and Prediction. Subsequently, for the convolutional module, 1d-max pooling is applied to **c**, which ought to capture the most prominent signals. In the memory module, the final sentiment vector is modulated by a weight matrix $\mathbf{W}_{\mathrm{s}}$ 2 $\mathsf{R}^{l \quad d_{\mathrm{s}}}$ to form a feature vector of dimensionality l. In general, we can use multiple filters to obtain several features in the convolutional module, while the memory module allows for adjusting the number of passes over the memory.

Finally, the outputs of these two modules are concatenated to form a fixed-length vector, which is passed to a fully connected softmax layer to obtain the final output probabilities.

Loss Function and Training. Our loss function is the cross-entropy function

$$L = \frac{1}{n} \sum_{i=1}^{n} \sum_{c2C} y_{i,c} \ln \hat{y}_{i,c}, \tag{3.6}$$

where n is the number of training examples, C is the set of (two) classes, $y_{i,c}$ are ground truth labels for a given training example and class c, and $\hat{y}_{i,c}$ are corresponding label probabilities predicted by the model, as emitted by the softmax layer. We train our model using Adam optimization [73] for better robustness across different datasets. Further details about our training regime follow in the Experiments section.

3.5 Experiments

We now turn to our extensive empirical evaluation, which assesses the effectiveness of our novel architecture with sentiment word vectors.

3.5.1 Experimental Setup

Datasets. For evaluation, we use real world datasets for 7 different languages, taken from a range of different sources that cover several domains. These are summarized in Table 3.1,

Language	Source	Domain	train	test
en	SST	Movies	6,920	1,821
	AFF	Food	5,945	1,189
es	SE16-T5	Restaurants	2,070	881
ru	TA	Hotels	2,387	682
de	TA	Restaurants	1,687	481
cs	TA	Restaurants	1,722	491
it	TA	Hotels	3,437	982
ja	TA	Restaurants	1,435	411

Table 3.1: Sentiment Dataset Descriptions

with ISO 639-3 language codes. In our experimental setup, these are all cast as binary polarity classification tasks, for which we use accuracy as our evaluation metric.

- The Stanford Sentiment Treebank (**SST**) dataset [9] consists of movie reviews taken from the Rotten Tomatoes website, including binary labels. We only used sentence-level data in our experiment.
- The SemEval-2016 Task 5 (**SE16-T5**) dataset [74] provides Spanish reviews of restaurants. It targeted aspect-based sentiment analysis, so we converted the entity-level annotations to sentence-level polarity labels via voting. As the number of entities per sentence is often one or very low, this process is reasonably precise. In any case, it enables us to compare the ability of different model variants to learn to recognize pertinent words.
- From TripAdvisor (**TA**), we crawled German, Russian, Italian, Czech, and Japanese reviews of restaurants and hotels. We removed three-star reviews, as these can be regarded as neutral ones, so reviews with a rating < 3 are considered negative, while those with a rating > 3 were deemed positive.
- The Amazon Fine Food Reviews **AFF** [75] dataset provides food reviews left on Amazon. We chose a random subset of it with preprocessing as for TripAdvisor.

As there was no test set provided for TripAdvisor or for the Amazon Fine Food Reviews data, we randomly partitioned this data into training, validation, and test splits with a 80%/10%/20% ratio. Additionally, 10% of the training sets from SE16-T5 were randomly

Description		Values
Convol. Module	filter region size	(3,4,5)
	feature maps	100
	pooling	1d-max pooling
Memory Module	# passes (k)	2
	feature vector size	100
dropout rate		0.5
optimizer		Adam
activation function		ReLU
batch size		50

Table 3.2: DM-MCNN Model Parameter Settings.

extracted and reserved for validation, while SST provides its own validation set. The new datasets are available from http://gerard.demelo.org/sentiment/.

Embeddings. The standard pre-trained word vectors used for English are the GloVe [69] ones trained on 840 billion tokens of Common Crawl data[1], while for other languages, we rely on the Facebook fastText Wikipedia embeddings [76] as input representations. All of these are 300-dimensional. The vectors are either fed to the CNN, or to the convolutional module of the DM-MCNN during initialization, while unknown words are initialized with zeros. All words, including the unknown ones, are fine-tuned during the training process.

For our transfer learning approach, our experiments rely on the multi-domain sentiment dataset by Blitzer *et al.* [77], collected from Amazon customers reviews. This dataset includes 25 categories of products and is used to generate our sentiment embeddings using linear models. Specifically, we train linear SVMs using scikit-learn to extract word coefficients in each domain and also for the union of all domains together, yielding a 26-dimensional sentiment embedding.

For comparison and analysis, we also consider several alternative forms of infusing external cues. Firstly, lexicon-driven methods have often been used for domain-independent sentiment analysis. We consider a recent sentiment lexicon called VADER [18]. The polarity scores assigned to words by the lexicon are taken as the components of a set of 1-dimensional

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

word vectors (dividing the original scores by the difference between max and min polarity scores for normalization). Secondly, as another particularly strong alternative, we consider the SocialSent Reddit community-specific lexicons mined by the Stanford NLP group [19]. These contain separate domain-specific scores for 250 different Reddit communities, and hence result in 250-dimensional embeddings.

For cross-lingual projection, we extract links between words from a 2017 dump of the English edition of Wiktionary. We restrict the vocabulary link set to include the languages in Table 3.1, mining corresponding translation, synonymy, derivation, and etymological links from Wiktionary.

Neural Network Details. For CNNs, we make use of the well-known CNN-non-static architecture and hyperparameters proposed by Kim [10], with a learning rate of 0.0006, obtained by tuning on the validation data. For our DM-MCNN models, the configuration of the convolutional module is the same as for CNNs, and the remaining hyperparameter values were as well tuned on the validation sets. An overview of the relevant network parameter values is given in Table 3.2.

For greater efficiency and better convergence properties, the training relies on mini-batches. Our implementation considers the maximal sentence length in each mini-batch and zero-pads all other sentences to this length under convolutional module, thus enabling uniform and fast processing of each mini-batch. All neural network architectures are implemented using the PyTorch framework[2].

3.5.2 Results and Analysis

Baseline Results. Our main results are summarized in Table 3.3. We compare both regular CNNs and our dual-module alternative DM-MCNNs under a variety of settings. A common approach is to use a CNN with randomly initialized word vectors. Comparing this to CNNs with GloVe/fastText embeddings, where GloVe is used for English, and fastText is used for

[2]http://pytorch.org

Approach	d	en		es	ru	de	cs	it	ja
		Movies	Food	Rest.	Hotels	Rest.	Rest.	Hotels	Rest.
CNN									
— Random Init.	300	80.78	86.63	81.50	90.18	88.09	90.00	93.18	78.59
— Word Vec. Init.	300	85.72	87.97	85.13	92.82	92.10	92.46	96.20	77.62
Our Approach									
— With fine-tuning	300/26	**86.99**	**90.08**	85.02	93.40	93.14	93.08	95.50	**85.40**
— No fine-tuning	300/26	86.38	88.81	**85.70**	**94.87**	**94.59**	**93.48**	96.20	77.62
CNN with Concatenated Sentiment Embeddings									
— VADER	301	85.89	88.39	84.90	92.31	88.36	93.08	96.34	77.62
— SocialSent	550	84.90	88.48	82.63	92.23	91.48	86.56	94.51	76.64
— Our Embeddings	326	86.05	89.07	84.56	92.72	93.56	91.24	95.78	77.62
Our Model with Alternative Sentiment Embeddings									
— Random	300/26	86.16	87.97	85.24	93.99	93.14	92.67	96.20	80.29
— VADER	300/1	86.33	88.39	84.45	94.18	92.31	92.87	96.48	75.43
— SocialSent	300/250	86.38	87.89	83.09	93.40	92.31	93.28	**96.62**	81.02

Table 3.3: Accuracy on several different English and non-English datasets from different domains, comparing our architecture against CNNs. Rest.: restaurants domain.

all other languages, we observe substantial improvements across all datasets. This shows that word vectors do tend to convey pertinent word semantics signals that enable models to generalize better. Note also that the accuracy using GloVe on the English movies review dataset is consistent with numbers reported in previous work [78].

Dual-Module Architecture. Next, we consider our DM-MCNNs with their dual-module mechanism to take advantage of transfer learning. We observe fairly consistent and sometimes quite substantial gains over CNNs with just the GloVe/fastText vectors. We see that the sentiment embeddings provide important complementary signals beyond what is provided in

regular word embeddings, and that our dual-module approach succeeds at exploiting these signals across a range of different domains and languages. Our transfer learning approach leads to sentiment embeddings that capture signals from multiple domains. The model successfully picks the pertinent parts of this signal for datasets from domains as different as movie reviews and food reviews.

We report results for two different training conditions. In the first condition (with fine-tuning), the sentiment embedding matrix is preinitialized using the data from our transfer learning procedure, but the model is then able to modify these arbitrarily via backpropagation. In the second condition (no fine-tuning), we simply use our sentiment embedding matrix as is, and do not update it. Instead, the model is able to update its various other parameters, particularly its various weight matrices and bias vectors. While both training conditions outperform the CNN baseline, there is no obvious winner among the two. When the training data set is very small and hence there is a significant risk of overfitting, one may be best advised to forgo fine-tuning. In contrast, when it is somewhat larger (as for our English datasets, which each have over 5,000 training instances) or when the language is particularly idiosyncratic or not covered sufficiently well by our cross-lingual projection procedure (such as perhaps for Japanese), then fine-tuning is recommended. In this case, fine-tuning may allow the model to adjust the embeddings to cater to domain-specific meanings and corpus-specific correlations, while also overcoming possible sparsity of the cross-lingual vectors resulting from a lack of coverage of the translation dictionary.

It is important to note that many of the results in Table 3.3 stem from embeddings that were created automatically using cross-lingual projection. Our transfer learning embeddings were induced from entirely English data. Although the automatically projected cross-lingual embeddings are very noisy and limited in their coverage, particularly with respect to inflected forms, our model succeeds in exploiting them to obtain substantial gains in several different languages and domains.

Alternative Embedding Methods. For a more detailed analysis, we conducted additional

experiments with alternative embedding conditions. In particular, as a simpler means of achieving gains over standard CNNs, we propose to use CNNs with word vectors augmented with sentiment cues. Given that regular word embeddings appear to be useful for capturing semantics, one may conjecture that extending these word vectors with additional dimensions to capture sentiment information can lead to improved results. For this, we simply concatenate the regular word embeddings with different forms of sentiment embeddings that we have obtained, including those from the sentiment lexicon VADER, from the Stanford SocialSent project, and from our transfer learning procedure via Amazon reviews. To conduct these experiments, we also produced cross-lingual projections of the VADER and SocialSent embedding data.

The results of using these embeddings as opposed to regular ones are somewhat mixed. Concatenating the VADER embeddings or our transfer learning ones leads to minor improvements on English, and our cross-lingual projection of them leads to occasional gains, but the results are far from consistent. Even on English, adding the 250-dimensional SocialSent embedding seems to degrade the effectiveness of the CNN, although all input information that was previously there continues to be provided to it. This suggests that a simple concatenation may harm the model's ability to harness the semantic information carried by regular word vectors. This risk seems more pronounced for larger-dimensional sentiment embeddings.

In contrast, with our DM-MCNNs approach, the sentiment information is provided to the model in a separate memory module that makes multiple passes over this data before combining it with the regular CNN module's signals. Thus, the model can exploit the two kinds of information independently, and learn a suitable way to aggregate them to produce an overall output classification.

This hence demonstrates not only that the sentiment embeddings tend to provide important complementary signals but also that a dual-module approach is best-suited to incorporate such signals into deep neural models.

We also analysed our DM-MCNNs with alternative embeddings. When we feed random sentiment embeddings into them, not unexpectedly, in many cases the results do not improve much. This is because our memory module has been designed to leverage informative prior information and to re-weight its signals based on this assumption. Hence, it is important to feed genuine sentiment cues into the memory module. Yet, on some languages, we nevertheless note improvements over the CNN baseline. In these cases, even if similarities between pairs of sentiment vectors initially do not carry any significance, backpropagation may have succeeded in updating the sentiment embedding matrix such that eventually the memory module becomes able to discern salient patterns in the data.

We also considered our DM-MCNNs when feeding the VADER or SocialSent embeddings into the memory module. In this case, it also mostly succeeded in outperforming the CNN baseline. In fact, on the Italian TripAdvisor dataset, the SocialSent embeddings yielded the overall strongest results. In all other cases, however, our transfer learning embeddings proved more effective. We believe that this is because they are obtained in a data-driven manner based on an objective that directly seeks to optimize for classification accuracy.

Model	*20%*	*50%*	*100%*
CNN [10]	83.14	84.29	85.72
CNN-Rule-q [79]	83.75	85.45	86.49
Gumbel Tree-LSTM [80]	84.04	84.83	86.80
DC-MCNN (ours)	**85.06**	**86.16**	**86.99**

Table 3.4: Accuracy on SST with increasing training sizes

Influence of Training Set Size. To look into the effect of our approach with restricted training data, we first consider the SST dataset as an instructive example. We set the training set size to 20%, 50%, 100% of its original size and compared our full dual module model with sentiment embeddings against state-of-the-art methods.

The results are given in Table 3.4. Our dual module CNN has a sizeable lead over other methods when only using 20% of SST training set. Given that we study how to incorporate extrinsic cues into a deep neural model, we consider CNN-Rule-q [79] and Gumbel Tree-

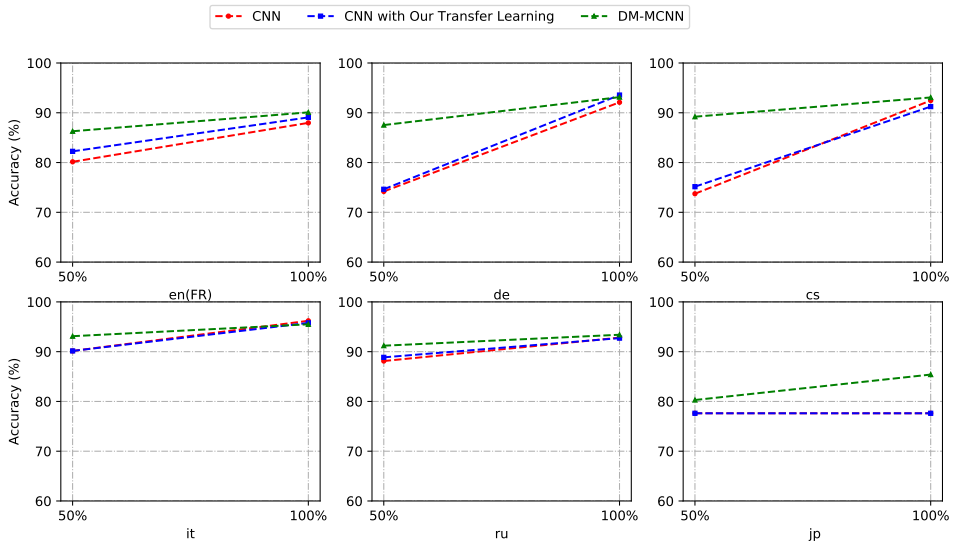

Figure 3.2: Effectiveness of three embedding alternatives on 6 languages at a reduced training size (comparing 50% and 100%).

LSTM [80] as the relevant baseline methods. The CNN-Rule-q method used an iterative distillation method that exploits structured information from logical rules, which for SST is based on the word *but* to determine the weights in the neural network. The Gumbel Tree-LSTM approach incorporates a Straight-Through Gumbel-Softmax into a tree-structured LSTM architecture that learns how to compose task-specific tree structures starting from plain raw text. They all require a large amount of data to pick up sufficient information during training, while our method is able to efficiently capture sentiment information from our transfer learning even though the data is scarce.

For further analysis, we also artificially reduce the training set sizes to 50% of the original sizes given in Table 3.1 for our multilingual datasets. The results are plotted in Figure 3.2. We compare: 1) the CNN model baseline, 2) the CNN model but concatenating our sentiment embeddings from transfer learning, and 3) our full dual module model with these sentiment embeddings. We already saw in Table 3.3 that we obtain reasonable gains over generic embeddings when using the full training sets.

In Figure 3.2, we additionally observe that the gains are overall much more remarkable on smaller training sets. This shows that the sentiment embeddings are most useful when

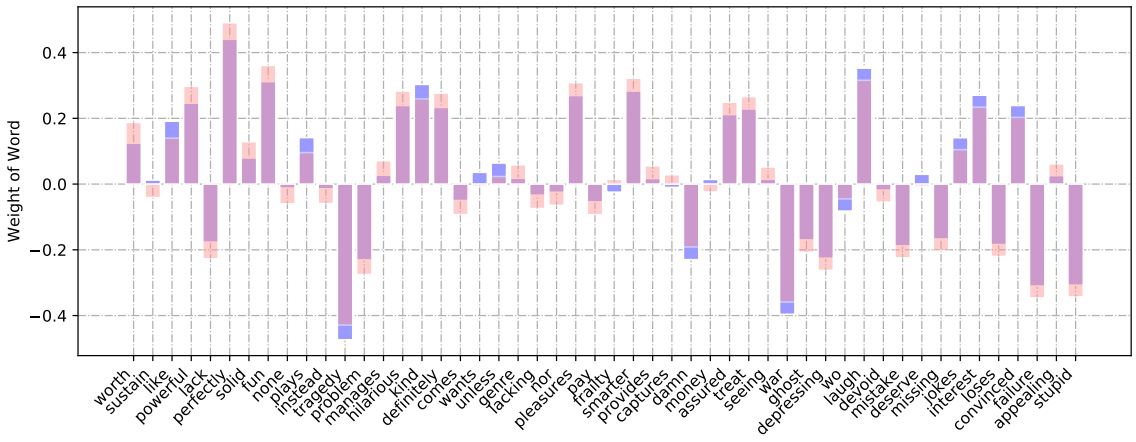

Figure 3.3: Top 50 weight changes of words fine-tuned by the sentiment memory module of the DM-MCNN, using the one-dimensional VADER embeddings, but considering only words with non-zero values in the original VADER data. Here, the dark shade (blue) refers to the original value of word vectors, while the light shade (red) refers to their fine-tuned values after training. The medium intensity (purple) corresponds to the overlap between the original and fine-tuned word vectors.

they are of high quality and domain-specific training data is scarce, although a modest amount of training data is still needed for the model to be able to adapt to the target domain.

Inspection of the DM-MCNN-learned Deep Sentiment Information. To further investigate what the model is learning, we examine the changes of weights of words on the English SST dataset when using the VADER sentiment embeddings with DM-MCNNs. Although these are not as powerful as our transfer learning embeddings, the VADER embeddings are the most easily interpretable here, since they are one-dimensional, and thus can be regarded as word-specific weights. The result is visualized in Figure 3.3. Here, the dark-shaded segments (in blue) refer to the original weights, while the light-shaded segments (in red) refer to the adjusted weights after training. The medium-shaded segments (in purple) reflect the overlap between the two. Hence, whenever we observe a dark (blue) segment above a medium (purple) segment in a bar, we can infer that the fine-tuned weight for a word (e.g., for "*plays*" in Figure 3.3) was lower than the original weight of that word. Conversely, whenever we observe a light (red) segment at the top, the weight increased during training (e.g., for *hilarious*). Generally, dark (blue) segments reflect decreased weight magnitudes

and light (red) ones reflect increased weight magnitudes, both on the positive and on the negative side.

We consider in Figure 3.3 the top 50 weight changes only of words that were already covered by the original VADER sentiment embeddings. Here, it is worth noting that the weight magnitudes of positive words such as "*laugh*", "*appealing*" and negative words such as "*lack*", "*missing*" increase further, while words such as "*damn*", "*interest*", "*war*" see decreases in magnitude, presumably due to their ambiguity and context (e.g., "*damn good*", "*lost the interest*", descriptions of war movies). Hence, the figure confirms that our DM-MCNN approach is able to exploit and customize the provided sentiment weights for the target domain. However, unlike the VADER data, our transfer learning approach results in multi-dimensional sentiment embeddings that can more easily capture multiple domains right from the start, thus making it possible to use them even without further fine-tuning.

3.6 Discussion

Deep neural networks are widely used in sentiment polarity classification, but suffer from their dependence on very large annotated training corpora. In this chapter, we show how to build multilingual sentiment embeddings and then study how to incorporate them as extrinsic cues into the network, beyond just generic word embeddings. We have found that this is best achieved using a dual-module approach that encourages the learning of models with favourable generalization abilities. Our experiments show that this can lead to gains across a number of different languages and domains.

CHAPTER 4
SELF-LEARNING IN CROSS-LINGUAL TEXT CLASSIFICATION

4.1 Overview

In this chapter, we propose a self-learning framework to incorporate the predictions of the multilingual BERT model [2] on non-English data into an English training procedure. The initial multilingual BERT model was simultaneously pretrained on 104 languages, and has shown to perform well for cross-lingual transfer of natural language tasks [81]. Our model begins by learning just from available English samples, but then makes predictions on unlabeled non-English samples and a part of those samples with high confidence prediction scores are repurposed to serve as labeled examples for a next iteration of fine-tuning until the model converges. During each iteration, the resulting model serves as a teacher to induce labels for unlabeled target language samples that can be used during further adversarial training, allowing us to gradually adapt our model to the target language.

Based on this multilingual self-learning technique, we demonstrate the superiority of our framework on Multilingual Document Classification (MLDoc) [6] and Cross-lingual Intent Classification [7] in comparison with several strong baselines. Our study then proceeds to show that our method is better on Chinese sentiment classification than other cross-lingual methods that also consider unlabeled non-English data. This shows that our method is more effective at cross-lingual transfer for domain-specific tasks, using a mix of labeled and unlabeled data via a multilingual BERT sentence model.

4.2 Related Work

Semi-supervised Learning. There is a long history of research on semi-supervised Learning to exploit unlabeled data. Self-learning (also known as self-training) was successfully applied

to NLP tasks in early work such as on word sense disambiguation [82] and parsing [83]. In recent work, Artetxe *et al.* [84] show that self-learning can iteratively improve unsupervised cross-lingual word embeddings. Clark *et al.* [85] presents Cross-View Training, a new self-training algorithm that works well for neural sequence modeling. Other semi-supervised methods, such as co-training [86] and tri-training [87], have as well been used for sentiment analysis. Ruder and Plank [88] propose a novel multi-task tri-training method that reduces the time and space complexity of classic tri-training for sentiment analysis. For cross-lingual sentiment analysis, Wan [89] uses machine translation to directly convert English training data to Chinese, which provides two views for co-training. Xu and Yang [90] propose to use soft probabilistic predictions for the documents in a label-rich language as the (induced) supervisory labels in a parallel corpus of documents, while there is no need to use parallel corpora in our work. Chen *et al.* [91] propose an Adversarial Deep Averaging Network to learn invariance across languages, which is another baseline considered in our experiments.

Adversarial Training. There is substantial research on learning to resist adversarial perturbations with the goal of improving the robustness of a machine learning system [92, 93, 94]. In natural language processing, adversarial perturbation has proven effective for improving a model's generalization [95, 96, 97, 98]. Miyato *et al.* [95] adopt adversarial and virtual adversarial training for improved semi-supervised text classification in monolingual settings. Fu *et al.* [99] propose an Adversarial Bi-directional Sentence Embedding Mapping framework to learn cross-lingual mappings of sentence representations. Cheng *et al.* [97] improve a translation model with adversarial source examples. In our experiments, we show that our approach outperforms adversarial perturbation applied to Multilingual BERT.

Cross-lingual Representation Learning. With models such as ELMo [20], GPT-2 [21], and BERT [2], important progress has been made in learning improved sentence representations with context-specific encodings of words via a language modeling objective. The latter two approaches both rely on Transformer encoders, but BERT is trained using masked language modeling instead of right-to-left or left-to-right language modeling. Additionally, BERT

also optimizes a next sentence classification objective. Recent work has also investigated cross-lingual extensions. Devlin *et al.* [2] themselves published a multilingual version of BERT, following the same model architecture and training procedure, except that the union of 104 different language editions of Wikipedia serves as the training input. Lample and Conneau [5] incorporate parallel text into BERT's architecture by training on a new supervised learning objective. Artetxe and Schwenk [4] also show that the encoder from a pretrained sequence-to-sequence model can be used to produce cross-lingual sentence embeddings. All these methods are compatible with our self-learning framework, since they provide a shared sentence meaning representation across languages as needed by our approach.

4.3 Self-Learning

Our proposed framework consists of three parts as shown in Figure 4.2. The first part is the pretrained multilingual encoder, denoted as $f_n(\ ;\theta_n)$. The encoder is assumed to have been pretrained across different languages with appropriate strategies, such as WordPiece, Byte Pair Encoding modeling, and Cross-lingual Word Alignment, to allow the model to share representations across languages to a certain degree. Hence, we obtain a universal sentence representation $\mathbf{h} \in \mathsf{R}^d$ from this encoder, where d is the dimensionality of the sentence representation. Subsequently, a task-specific classification module $f_{\mathrm{cl}}(\ ;\theta_{\mathrm{cl}})$ is applied for fine-tuning on top of the pretrained model $f_n(\ ;\theta_n)$. This module consists of a linear function mapping $\mathbf{h} \in \mathsf{R}^d$ into $\mathsf{R}^{|Y|}$ and a softmax function, where Y is the set of target classes.

For the overall process, we first train the whole network $f(\ ;\theta)$ in K epochs using a set of the labeled data $L = \{(x_i, y_i) \mid i = 1, ..., n\}$, where n is the number of labeled instances, $x_i \in X$ are instances, and $y_i \in Y$ are the corresponding ground truth labels. The next step is to make predictions for the unlabeled instances in $U = \{x_u \mid u = 1, ..., m\}$. We assume that $f(\ ;\theta)$ yields a class label as well as a confidence score.

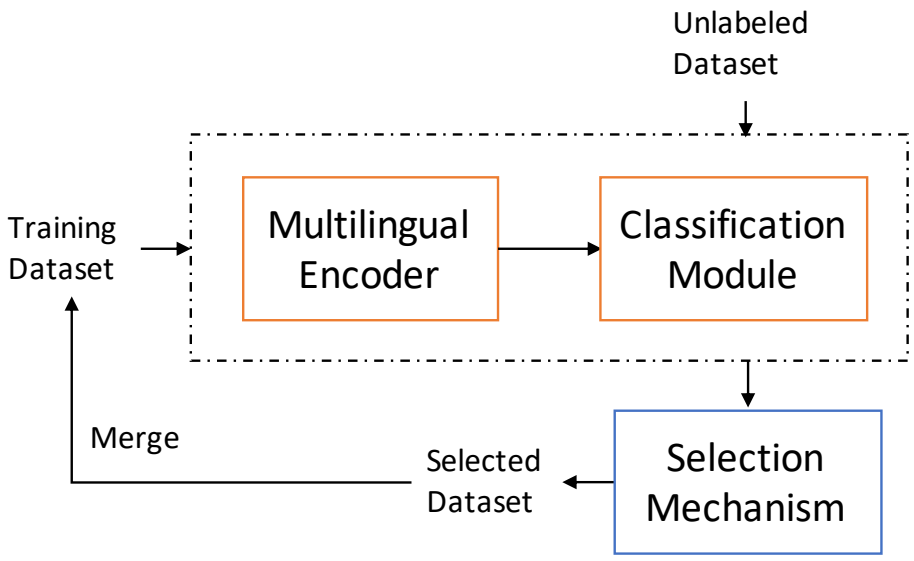

Figure 4.1: Illustration of self-learning process for cross-lingual classification.

To better take advantage of the pretrained multilingual model, a selection mechanism is invoked to repurpose unlabeled data with high confidence scores for incorporation into the training data. There are several variations of such selection mechanisms [100], and we rely on a *balancing* approach that considers the same number of instances for each class. Thus, we select a subset $\{x_s \mid s = 1, ..., K_t\}$ of the unlabeled data for each class, containing the top K_t highest confidence items based on the current trained model. The union set U_s of selected items is merged into the training set L and then we retrain the model and repeat this process iteratively. The detailed process is described in Algorithm Algorithm 1, where for each class $y \in Y$, D_y denotes the set of tuples pairing unlabeled data with corresponding confidence scores c.

4.4 Adversarial Training in Self-Learning

Still, there may still be subtle differences between languages that are neglected when doing so. To address this, we further present a semi-supervised adversarial training process that minimizes the maximal loss for label-preserving input perturbations. The resulting model then serves as a teacher to induce labels for unlabeled target language samples that can be

Algorithm 1 Self-learning on cross-lingual tasks

repeat
  Fine-tune $f(\ ;\theta)$ for K epochs using L
  for y 2 Y **do**
    D_y ;
  for x_{u} 2 U **do**
    (y, c) $f(x_u; \theta)$
    D_y D_y [$f(x_{\mathrm{u}}, c)g$
  for y 2 Y **do**
    S_y $\underset{S \ D_y,\ jSj \ K_t}{\operatorname{argmax}} \sum_{(x_{\mathrm{u}},c)2S} c$
    U U n S_y
    U_{s} U_{s} [S_y
  L L [U_{s}
until stopping criterion is true

used during further adversarial training, allowing us to gradually adapt our model to the target language.

Overview of the Method. Our proposed method consists of two parts, as illustrated in Figure 4.2. The backbone is a multilingual classifier, which includes a pretrained multilingual encoder $f_n(\ ;\theta_n)$ and a task-specific classification module $f_{\mathrm{cl}}(\ ;\theta_{\mathrm{cl}})$. By adopting an encoder that (to a certain degree) shares representations across languages, we obtain a universal text representation $\mathbf{h}$ 2 R^d, where d is the dimensionality of the text representation. The classification module $f_{\mathrm{cl}}(\ ;\theta_{\mathrm{cl}})$ is applied for fine-tuning on top of the pretrained model $f_n(\ ;\theta_n)$. It applies a linear function to map $\mathbf{h}$ 2 R^d into R^{jYj}, and a softmax function, where Y is the set of target classes.

Adversarial Training. Our adversarial self-learning process proceeds as follows. First, we train the entire network $f(\ ;\theta)$ in K epochs using a set of labeled data $L = f(\mathbf{x}_i, y_i)$ j $i = 1, ..., ng$ from the source language, where n is the number of labeled instances, $\mathbf{x}_i$ 2 X consists of embedding vectors $[\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_T]$ for each instance (T is the length of one sequence), and y_i 2 Y are the corresponding ground truth labels.

Adversarial training is motivated by the idea of making the model robust against adver-

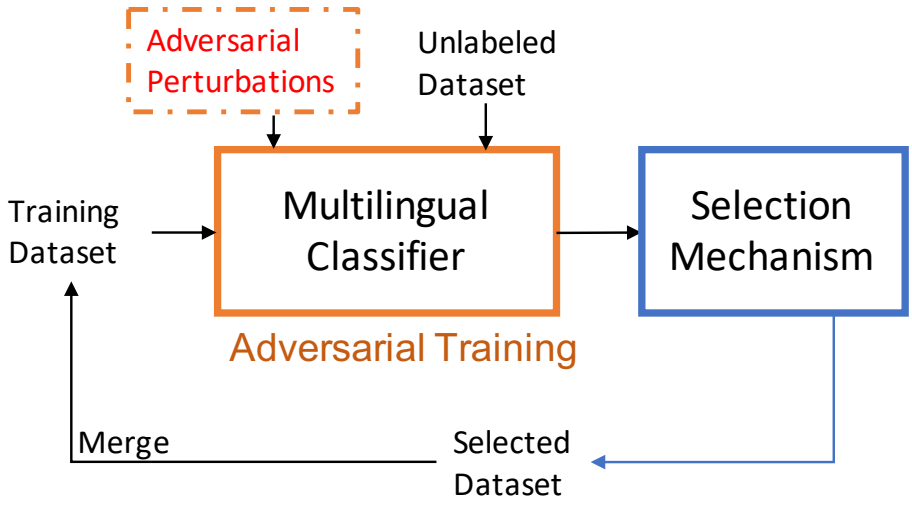

Figure 4.2: Illustration of self-learning process with adversarial training for cross-lingual classification.

sarial examples. It is well-known that deep neural networks are susceptible to perturbed inputs that have been deliberately constructed to fool the network into making a misclassification [92]. Adversarial training is based on the notion of making the model robust against such perturbation, i.e., against an imagined adversary that seeks out minuscule changes to an input that lead to a misclassification, assuming that the class label should not actually be affected by such minuscule changes. To perform adversarial training, the loss function becomes:

$$L_{\text{adv}}(\mathbf{x}_i, y_i) = L(f(\mathbf{x}_i + \mathbf{r}_{\text{adv}}; \theta), y_i) \tag{4.1}$$

$$\text{where } \mathbf{r}_{\text{adv}} = \underset{\mathbf{r}, jjrjj \ \epsilon}{\text{argmax}} L(f(\mathbf{x}_i + \mathbf{r}; \tilde{\theta}), y_i).$$

Here $\mathbf{r}$ is a perturbation on the input and $\tilde{\theta}$ is a set of parameters set to match the current parameters of the entire network, but ensuring that gradient propagation only proceeds through the adversarial example construction process. At each step of training, the worst case perturbations $\mathbf{r}_{\text{adv}}$ are calculated against the current model $f(\mathbf{x}_i; \tilde{\theta})$ in Equation 4.1, and we train the model to be robust to such perturbations by minimizing Equation 4.1 with

respect to θ. We later empirically confirm that adding random noise instead of seeking such adversarial worst case perturbations is not able to bring about similar gains.

Generally, we cannot obtain a closed form for the exact perturbation $\mathbf{r}_{\text{adv}}$, but Goodfellow *et al.* [93] proposed to approximate this worst case perturbation $\mathbf{r}_{\text{adv}}$ by linearizing $f(\mathbf{x}_i; \tilde{\theta})$ around $\mathbf{x}_i$. With a linear approximation and an L_2 norm constraint in Equation 4.2, the adversarial perturbation is

$$\mathbf{r}_{\text{adv}} \quad \epsilon \frac{\mathbf{g}}{jjgjj_2} \tag{4.2}$$
$$\text{where } \mathbf{g} = r_{\mathbf{x}_i} L(f(\mathbf{x}_i; \tilde{\theta}), y_i)$$

During the actual training, we optimize the loss function of the adversarial training in Equation 4.1 based on the adversarial perturbation defined by Equation 4.2 in each step.

4.5 Experiments

We evaluate our self-learning framework on two cross-lingual document and sentiment classification tasks to show the effectiveness of self-learning for multilingual BERT-based cross-lingual transfer.

4.5.1 Experimental Setup

Datasets. For evaluation, we first rely on MLDoc [6], a balanced subset of the Reuters corpus covering 8 languages for document classification, with 1,000 training and validation documents and 4,000 test documents for each language. The 4-way topic labeling consists of *Corporate*, *Economics*, *Government*, and *Market*. For cross-lingual classification, 1,000 target language training documents serve as unlabeled data for self-learning.

We further evaluate our framework on cross-lingual intent classification from English to Spanish and Thai [7]. This dataset is built for multilingual task oriented dialogue. It contains 57k annotated utterances in English (43k), Spanish (8.6k), and Thai (5k) with 12

Parameter	MLDoc	CLIC
max. sequence length	96	32
batch size	64	128
learning rate	2e-5	2e-6
K_{t}	50	30
# of training epochs	5	6
ϵ	1.0	1.0

Table 4.1: Hyper-parameters for our self-learning framework.

different intents across the domains *weather*, *alarms*, and *reminders*. 3k Spanish or 2k Thai training utterances are used as unlabeled data for self-learning. All classification tasks are evaluated in terms of classification accuracy (ACC).

We also evaluate our self-learning method on cross-lingual sentiment classification from English to Chinese. For English, we use a balanced dataset of 700k Yelp reviews from Zhang *et al.* [101] with their ratings as labels (scale 1–5) and adopting their training–validation split: 650k reviews for training and 50k as a validation set. For Chinese, we use the same dataset configuration as Chen *et al.* [91], consisting of 150k unlabeled Chinese hotel reviews and 10k balanced Chinese hotel reviews as a validation set. The results are reported on a separate test set of another 10k hotel reviews. The data are also annotated with 5 labels (1–5). Both classification tasks are evaluated in terms of classification accuracy (ACC).

Model Details. We tune the hyper-parameters for our neural network architecture based on each non-English validation set. For the multilingual encoder, we invoke the Multilingual BERT model [2], which supports 104 languages. Most hyper-parameters are shown in Table 4.1, with the exception that lower-casing is omitted for Thai and ϵ is 10 in the Japanese experiment. We rely on early stopping as a termination criterion, specifically, when the performance on the validation set stops improving in 2 self-learning iterations.

4.5.2 Results and Analysis

Cross-lingual Document Classification. Our MLDoc experiments compare our approach against several strong baselines. Schwenk and Li [6] propose MultiCCA, consisting of

Approach	ACC
Domain Adaptation	
mSDA [102]	31.44
Machine Translation	
DAN + MT [91]	39.66
CLD-based CLTC	
CLD-KCNN [90]	40.96
CLDFA-KCNN [90]	41.82
Cross-lingual Adversarial	
ADAN [91]	42.49
Cross-lingual transfer	
BERT	40.73
Our Approach	**43.88**

Table 4.2: Accuracy (in %) on Chinese sentiment classification without using labeled Chinese data. CLD and CLTC represent cross-lingual distillation and cross-lingual text classification.

multilingual word embeddings and convolutional neural networks. Artetxe and Schwenk [4] pretrain a massively multilingual sequence-to-sequence neural MT model, invoking its encoder as a multilingual text representation used for fine-tuning on downstream tasks. Keung *et al.* [103] apply language-adversarial learning into Multilingual BERT during fine-tuning with unlabeled data. We also considered Multilingual BERT and itself with self-learning and adversarial training respectively as our baselines. Additionally, we compare Multilingual BERT with self-learning using unlabeled data as investigated by Dong and Melo [104]. As shown in Table 4.3, BERT with adversarial training outperforms Multilingual BERT across a range of languages, which establishes its merits for cross-lingual classification. Beyond this, our full framework further outperforms all baselines across 7 languages, including for phylogenetically unrelated languages.

Cross-lingual Intent Classification. To evaluate the generalization of our framework to cross-lingual intent classification, we consider a diverse set of baselines as listed in Table 4.4. Schuster *et al.* [7] propose to combine Multilingual CoVe [106] with an auto-encoder objective and then use the encoder with a CRF model. We also run experiments

Approach	en	de	zh	es	fr	it	ja	ru
In-language supervised learning								
Schwenk et al. (2018) [6]	92.2	93.7	87.3	94.5	92.1	85.6	85.4	85.7
BERT (2018)	94.2	93.3	89.3	95.7	93.4	88.0	88.4	87.5
Cross-lingual transfer								
— *Without unlabeled data*								
Schwenk et al. (2018) [6]	92.2	81.2	74.7	72.5	72.4	69.4	67.6	60.8
Artetxe et al. (2018) [4]	89.9	84.8	71.9	77.3	78.0	69.4	60.3	67.8
BERT	93.0	75.0	71.3	78.3	77.8	68.5	71.8	76.6
BERT + Random Perturbation	–	79.3	73.8	73.0	81.3	67.1	73.1	66.9
BERT + Adv. Perturbation	–	82.2	82.0	81.5	83.7	72.9	75.6	78.8
— *With unlabeled data*								
Keung et al. (2019) [103]	–	88.1	84.7	80.8	85.7	72.3	76.8	77.4
Dong et al. (2019) [104]	–	89.9	84.5	84.8	88.5	75.8	76.4	79.3
Our Approach w/ Random Perturbation	–	90.5	83.7	86.8	88.3	76.1	78.1	80.9
Our Approach w/ Adv. Perturbation	–	**91.8**	**86.7**	**90.0**	**89.9**	**78.9**	**78.7**	**83.3**

Table 4.3: Accuracy (in %) on MLDoc experiments. Bold denotes the best on cross-lingual transfer.

Approach	en	es	th
Schuster et al. (2018) [7]	99.11	53.89	70.70
Liu et al. (2019) [105]	–	90.20	73.43
BERT	99.20	82.42	62.77
BERT + Adversarial Training	99.23	87.87	67.20
BERT + Self-Learning	–	88.33	71.51
Our Approach	–	**92.41**	**75.95**

Table 4.4: Accuracy (in %) on cross-lingual intent classification without using labeled non-English data.

on Multilingual BERT and observe that it does not outperform the method from Liu *et al.* [105], because this method takes advantage of additional information by selecting 11 domain-related words as alignment seeds. However, our approach still achieves the new state-of-the-art result, which suggests that our adversarial framework for cross-lingual transfer is effective across different kinds of classification tasks.

Influence of Adversarial Perturbations. To further evaluate the effect of adversarial perturbation and straight-forwardly show that it enables robustness with respect to divergence in the test set, we conduct an additional experiment on code-switching data. We create

	en (0%/0%)	de (16%/52%)	zh (9%32%)	es (16%/52%)
BERT	93.0	83.5	87.5	87.4
+ Random Perturbation	92.4	86.7	87.9	88.4
+ Adv. Perturbation	**94.4**	**88.8**	**91.3**	**90.4**
	fr (15%/50%)	it (14%/49%)	ja (8%/35%)	ru (12%/44%)
BERT	86.4	87.8	85.5	84.0
+ Random Perturbation	87.8	89.1	86.3	85.5
+ Adv. Perturbation	**90.6**	**91.9**	**88.6**	**89.2**

Table 4.5: Accuracy (in %) on MLDoc English code-switching data. The respective ratios of replaced words from the vocabulary and replaced word token occurrences in the English test set are given in parentheses.

challenge datasets[1] that adopt the original English training data, while the test data consists of English documents in which we attempt to replace all vocabulary words with non-English translations based on the bilingual English to non-English dictionaries from MUSE[2]. As a result, the test set documents consist of a form of code-switched language, in which many words are non-English but the word order remains unchanged. The replacement rates are listed in Table 4.5, along with the experimental results. We observe that the baselines have a low accuracy when faced with such codeswitching in the test set. This applies to Multilingual BERT without perturbation as well as Multilingual BERT with random perturbation. In contrast, our adversarial perturbation is significantly more effective than no or random perturbation when dealing with this data, and it does not impede the accuracy on English compared with random noise, thus improving both generalization and robustness.

de	zh	es	fr	it	ja	ru
93%	92 %	89 %	92%	80%	84%	89%

Table 4.6: Percentages of instances added into the training set that are correct for the MLDoc data using our method.

Cross-lingual Sentiment Classification. To evaluate the robustness of our framework on cross-lingual sentiment classification, we consider several diverse baselines as listed in Table 4.2. mSDA [102] is a very effective method for cross-domain sentiment classification

[1]Publicly available at http://crosslingual.nlproc.org/

[2]https://github.com/facebookresearch/MUSE – We use a random but consistent choice in case there are multiple translations.

on Amazon reviews, which can also be used in cross-lingual tasks, but it has the worst performance. Deep Averaging Networks (DANs) by Iyyer *et al.* [107] consider an arithmetic mean of word vectors as a sentence representation and pass it to a classification module, while Chen *et al.* [91] translate the Chinese test text into English as a machine translation baseline. The third category of baselines includes Xu and Yang [90], who propose a cross-lingual distillation (CLD) method that makes use of soft source predictions on a parallel corpus to train a target model (CLD-KCNN). They further propose an improved variant (CLDFA-KCNN) that utilizes adversarial training for domain adaptation within a single language. Adversarial DAN (ADAN) by Chen *et al.* [91] is another state of the art baseline that improves cross-lingual generalization by means of adversarial training. We also run experiments on multilingual BERT and observe that it does not outperform CLD-based CLTC and ADAN, while our approach achieves the new state-of-the-art result, indicating that our self-learning method for cross-lingual transfer can be more effective than a diverse range of other approaches. In addition, we evaluate the proximity between incorrect prediction and the corresponding correct label in our sentiment task by means of mean squared error. The error of our method is 1.37, while for regular multilingual BERT it is 1.42, which also shows the superiority of our method.

Comparison of Selection Mechanisms. We ran experiments on two selection mechanisms. One is *balancing*, as described in section 4.3. An alternative is *throttling* by selecting the top n unlabeled examples without considering specific classes [100]. Our experiments on MLDoc show that the results suffer from a rapid decline during self-learning with *throttling*. This is because selecting from all samples leads to an imbalance between different classes and due to repeated error amplification this means that samples are increasingly likely to be assigned to the majority class in each self-learning iteration.

4.6 Discussion

While multilingual encoders have enabled better cross-lingual learning, the obtained models often are not attuned to the subtle differences that a model may encounter when fed with documents in an entirely new language. To address this, this work proposes an adversarial perturbation framework that makes the model more robust and enables an iterative self-learning process that allows the model to gradually adapt to the target language. We achieve new state-of-the-art results on cross-lingual document and intent classification and demonstrate that adversarial perturbation is an effective method for improved classification accuracy without any labeled training data in the target language.

CHAPTER 5
DATA AUGMENTATION WITH ADVERSARIAL TRAINING FOR CROSS-LINGUAL NLI

5.1 Overview

For our study, we focus on natural language inference (NLI), i.e., classifying whether a premise sentence entails, contradicts, or is neutral with regard to a hypothesis sentence [108]. This is a useful building block for applications involving semantic understanding. However, the task is also very challenging, as it not only requires accounting for very subtle differences in meaning but also inferring presuppositions and implications that are not explicitly stated. Due to these intricate subtleties, zero-shot cross-lingual models are often fairly brittle, while obtaining in-language training data is fairly costly.

To boost the performance of cross-lingual models, an intuitive thought is to draw on unlabeled data from the target language so as to enable the model to better account for the specifics of that language, rather than just being fine-tuned on the source language. A natural way of exploiting unlabeled data is to consider standard semi-supervised learning methods that leverage a model's own predictions on unlabeled target language inputs [109]. However, this strategy fails when the predictions are too noisy to serve as reliable training signals. *data augmentation* hence is explored to circumvent this problem. The idea, widespread in computer vision and speech recognition, is to generate new training data from existing labeled data. For images, a common approach is to apply transformations such as rotation and flipping, as these typically preserve the original label assigned to an image [110]. For text, in contrast, data augmentation is more challenging, and straightforward techniques include simple operations on words within the original training sequences, such as synonym replacement, random insertion, random swapping, or random deletion [25]. In practice,

however, there are two notable problems. One is that the synthesized data from data augmentation techniques may as well be noisy and unreliable. Second, new examples may diverge from the distribution of the original data. On NLI, these problems are particularly pronounced, as the very nature of this task is to account for subtle differences between sentences. Modified versions of the original sentences may no longer have the same meaning and entailments. Hence, existing data augmentation techniques often fail to boost the result quality.

In this chapter, we propose a novel data augmentation scheme to synthesize controllable and much less noisy data for cross-lingual NLI. This augmentation consists of two parts. One serves to encourage language adaptation by means of reordering source language words based on word alignments to better cope with typological divergency between languages, denoted as Reorder Augmentation (RA). Another seeks to enrich the set of semantic relationships between a *premise* and pertinent *hypotheses*, denoted as Semantic Augmentation (SA). Both are achieved by learning corresponding sequence-to-sequence (Seq2Seq) models.

The resulting samples along with their new labels serve as an enriched training set for the final cross-lingual training. During this phase, we invoke a special adversarial training regimen that enables the model to better learn from such automatically induced training samples and transfer more information to the target languages while better bridging the gap between typologically distinct languages. Our empirical study demonstrates the necessity of incorporating adversarial training into training with synthetic samples and the superiority of our new augmentation method on cross-lingual Natural Language Inference [111]. Remarkably, our cross-lingual approach even outperforms in-language supervised learning.

5.2 Related Work

Adversarial Training. Many approaches for improving the robustness of a machine learning system against adversarial perturbations [112] have been advanced. Goodfellow *et al.* [113]

proposed a fast gradient method based on linear perturbation of non-linear models. Later, Madry *et al.* [94] presented PGD-based adversarial training through multiple projected gradient ascent steps to adversarially maximize the loss. In NLP, Belinkov and Bisk [114] exploited structure-invariant word manipulation and robust training on noisy texts for improved robustness. Iyyer *et al.* [115] proposed syntactically controlled paraphrase networks with back-translated data and used them to generate adversarial examples. Adversarial training also plays a role in improving a neural model's generalization. For instance, Cheng *et al.* [116] used adversarial source examples to improve a translation model. Dong *et al.* [117] exploit FGM-based adversarial training in self-learning for improved cross-lingual text classification. In our setting, we count on adversarial training in the word embedding space and show that PGD-based adversarial training remains effective when the adversarial perturbation is applied to noisy augmented examples.

Cross-Lingual Representation Learning. Representation learning approaches for natural language include static word embedding models such as word2vec [118], along with cross-lingual variants [119] or post-hoc alignment techniques [120]. In recent years, the Transformer architecture [121] has come to dominate the field, with models such as BERT [2] to obtain contextual embedding representations, and variants such as RoBERTa [122]. Google's T5 [123] is another Transformer-based model based on a encoder–decoder architecture for text generation, pretrained on multiple different tasks. It was quickly discovered that BERT-style models can learn multilingual representations simply by training them on the union of multiple languages, as exemplified in the mBERT model [2]. Similarly, XLM-R [3] follows RoBERTa's training procedure but is explicitly tuned for robust cross-lingual representation learning. Our work builds on these results but shows that incorporating additional data augmentation coupled with adversarial training allows such models to perform significantly better.

5.3 Data Augmentation with Adversarial Training

Our proposed method consists of two steps as shown in **??**. The first involves inducing training examples with two data augmentation models. Next, a task-specific classifier is trained on both the original and the newly generated training instances, with adversarial perturbation for improved robustness and generalization.

5.3.1 Reorder Augmentation Model

Reorder augmentation is based on the intuition of making a model more robust with respect to differences in word order typology. If our training examples consist entirely of instances from a language L_{S} with a fairly strict subject–verb–object (SVO) word order such as English, the model will be less well equipped to pay attention to subtle semantic differences between sentences from a target language L_{T} obeying subject–object–verb (SOV) order. To alleviate this problem, we can rely on auxiliary data to diversify the training data. For this, we obtain word alignments for unannotated bilingual parallel sentence pairs covering L_{S} and an auxiliary language L_{A} that need not be the same as L_{T}. We then reorder all source sentences to match the word order of L_{A} based on the alignments, and train a model to apply such reordering on the NLI training instances.

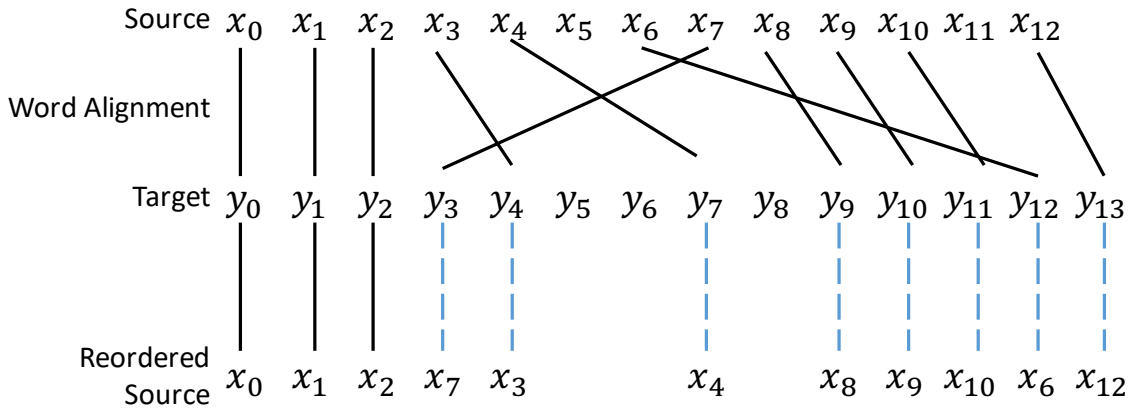

Figure 5.1: Illustration of using a word-aligned parallel corpus for reordering a source language text.

Formally, suppose we have obtained l unlabelled parallel sentences in the source language L_{S} and in the auxiliary language L_{A}, $C = \{(s_i, a_i) \mid i = 1, ..., l\}$, where (s, a) is a source–auxiliary language sentence pair. Based on a word alignment model, in our case *FastAlign* [124], which uses Expectation Maximization to compute the lexical translation probabilities, we obtain a word pair table for each sentence pair (s, t), denoted as $A(s, a) = \{(i_1, j_1), ..., (i_m, j_m)\}$.

Following the word order of L_{A}, we then reorder the source sequence s by consulting the table $A(s, t)$, yielding the new sentence pair $(s, \bar{s})$. Next, we consider a pretrained Seq2Seq model, denoted as $r(\ ; \theta)$. The model is assumed to have been pretrained with an encoder and a decoder in the source language, and we fine-tune this generative model by training on the new parallel corpus $\bar{C} = \{(s_i, \bar{s}_i) \mid i = 1, ..., l\}$. This generative Seq2Seq model can then reorder the sequences in the labeled training dataset $D = \{(\mathbf{x}_i, y_i) \mid i = 1, ..., n\}$, where n is the number of labeled instances, each $\mathbf{x}_i$ consists of a sequence pair (s_1, s_2), and each $y_i \in Y$ is the corresponding ground truth label describing their relationship.

5.3.2 Semantic Augmentation

Our second augmentation strategy involves training a controllable model that, given a sentence and a label describing the desired relationship, seeks to emit a second sentence that stands in said relationship to the input sentence. Thus, given an existing training sentence pair, we can consider different variations of one sentence in the pair and invoke the model to generate a suitable second sentence. However, such automatically induced samples from SA are inordinately noisy, precluding their immediate use as training data, so we exploit a large pretrained *Teacher model* trained on available source language samples to rectify the labels of these synthetic samples with appropriate strategies.

Generation. As we wish to be able to control the label of a generated example, the requested label is prepended to the input as a (textual) prefix before it is fed into a Seq2Seq

model. We adopt the ground-truth label of each example as the respective prefix, resulting in a new input sequence $(y_i : s_1)$ coupled with s_2 as the desired output forming a training pair for the generation model.

Given the resulting labeled training dataset D_{SA}, we can fine-tune a pretrained Seq2Seq model, denoted as $g(\ ;\theta)$. This generative Seq2Seq model can then be invoked for semantic data augmentation to generate new training instances. For each $(\bar{y} : s_1)$ as a labeled input sequence, where $\bar{y}$ 2 Y n f y_ig, we generate an $\tilde{s}_2$ via the fine-tuned Seq2Seq model, yielding a new training instance $(hs_1, \tilde{s}_2 i, \bar{y})$.

Label Rectification. The semantic augmentation induces $\tilde{s}_2$ automatically based on s_1 and the requested label $\bar{y}$. However, the obtained $\tilde{s}_2$ may not always genuinely have the desired relationship $\bar{y}$ to s_1. Thus, we treat this data as inherently noisy and propose a rectifying scheme based on a *Teacher* model. We wish for this *Teacher* to be as accurate as possible, so we start off with a large pretrained language model specifically for the source language L_{S}, which we assume obtains a better performance on L_{S} than a pretrained multilingual model. We train the Teacher network $h(\ ;\theta)$ in K epochs using the set of original labeled data D. This teacher model is then invoked to verify and potentially rectify labels from the automatically induced augmentation data $D_{\tilde{\mathrm{a}}} = f(\tilde{\mathbf{x}}_\mathbf{i}, y_i)\ j\ i = 1, ..., mg$ obtained in the previous step (where m is the number of instances). We assume $(\tilde{y}_i, c) = h(\tilde{\mathbf{x}}_\mathbf{i}; \theta)$ denotes the predicted label along with the confidence score c 2 $[0, 1]$ emitted by the classifier, and assume a confidence threshold T has been predetermined. There are several strategies to determine the final labels.

- **Teacher Strategy:** We adopt $D_{\mathrm{r}} = f(\tilde{\mathbf{x}}_\mathbf{i}, \tilde{y}_i)\ j\ (\tilde{\mathbf{x}}_\mathbf{i}, y_i)$ 2 $D_{\tilde{\mathrm{a}}}, (\tilde{y}_i, c) = h(\tilde{\mathbf{x}}_\mathbf{i}), c > Tg$, i.e., when the confidence score is above T, we believe the Teacher model is sufficiently confident to ensure a reliable label, while other instances are discarded.

- **TR Strategy:** An alternative scheme is to instead adopt $D_{\mathrm{r}} = f(\tilde{\mathbf{x}}_\mathbf{i}, \Phi(y_i, \tilde{y}_i, c))\ j$

$(\tilde{\mathbf{x}}_\mathbf{i}, y_i)$ 2 $D_{\tilde{\mathrm{a}}}, (\tilde{y_i}, c) = h(\tilde{\mathbf{x}}_\mathbf{i})g$, where

$$\Phi(y_i, \tilde{y_i}, c) = \begin{cases} \tilde{y_i} & c > T \\ y_i & \text{otherwise} \end{cases}$$

Here, labels remain unchanged when Teacher predictions match the originally requested labels. In case of an inconsistency, we adopt the Teacher model's label if it is sufficiently confident, and otherwise retain the requested label.

5.3.3 Projected Gradient Descent

Upon completing the two kinds of data augmentation, we possess synthesized data that is substantially less noisy, denoted as D_r, which can be incorporated into the original training data D to yield the final augmented training set $D_\mathrm{a} = D$ [D_r. With this, we proceed to train a new model $f(\ ;\theta)$ for the final cross-lingual sentence pair classification.

As a special training regimen, we adopt adversarial training, which seeks to minimize the maximal loss incurred by label-preserving adversarial perturbations [112, 113], thereby promising to make the model more robust. Nonetheless, the gains observed from it in practice have been somewhat limited in both monolingual and cross-lingual settings. We conjecture that this is because it has previously merely been invoked as an additional form of monolingual regularization [125].

In contrast, we hypothesize that adversarial training is particularly productive in a cross-lingual framework when used to exploit augmented data, as it encourages the model to be more robust towards the divergence among similar words and word orders in different languages and to better adapt to the new modestly noisy data. This hypothesis is later confirmed in our experimental results.

Adversarial training is based on the notion of finding optimal parameters θ to make the model robust against any perturbation $\mathbf{r}$ within a norm ball on a continuous multilingual

(sub-)word embedding space. Hence, the loss function becomes:

$$L_{\text{adv}}(\mathbf{x}_i, y_i) = L(f(\mathbf{x}_i + \mathbf{r}_{\text{adv}}(\mathbf{x}_i, y_i); \theta), y_i) \tag{5.1}$$
$$\text{where } \mathbf{r}_{\text{adv}}(\mathbf{x}_i, y_i) = \underset{\mathbf{r}, jjrjj \ \epsilon}{\text{argmax}} L(f(\mathbf{x}_i + \mathbf{r}; \tilde{\theta}), y_i)$$

Generally, a closed form for the optimal perturbation $\mathbf{r}_{\text{adv}}(\mathbf{x}_i, y_i)$ cannot be obtained for deep neural networks. Goodfellow *et al.* [113] proposed approximating this worst case perturbation by linearizing $f(\mathbf{x}_i; \tilde{\theta})$ around $\mathbf{x}_i$. With a linear approximation and an L_2 norm constraint in Equation 5.2, the adversarial perturbation is

$$\mathbf{r}_{\text{adv}}(\mathbf{x}_i, y_i) \quad \epsilon \frac{\mathbf{g}(\mathbf{x}_i, y_i)}{jj\mathbf{g}(\mathbf{x}_i, y_i)jj_2} \tag{5.2}$$
$$\text{where } \mathbf{g}(\mathbf{x}_i, y_i) = r_{\mathbf{x_i}} L(f(\mathbf{x}_i; \tilde{\theta}), y_i).$$

However, neural networks are typically not linear even over a relatively small region, so this approximation cannot guarantee to achieve the best optimal point within the bound. Madry *et al.* [94] demonstrated that projected gradient descent (PGD) allows us to find a better perturbation $\mathbf{r}_{\text{adv}}(\mathbf{x}_i, y_i)$. In particular, for the norm ball constraint $jj\mathbf{r}jj \quad \epsilon$, given a point r_0, $\Pi_{jj\mathbf{r}jj \ \epsilon}$ aims to find a perturbation $\mathbf{r}$ that is closest to r_0 as follows:

$$\Pi_{jj\mathbf{r}jj \ \epsilon}(r_0) = \underset{jj\mathbf{r}jj \ \epsilon}{\text{argmin}} jj\mathbf{r} \quad r_0 jj \tag{5.3}$$

To find more optimal points, K-step PGD is needed during training, which requires K forward–backward passes through the network. With a linear approximation and an L_2 norm constraint, PGD takes the following step in each iteration:

$$\mathbf{r}_{t+1} = \Pi_{jj\mathbf{r}jj \ \epsilon}\left(\mathbf{r}_t + \alpha \frac{\mathbf{g}(\mathbf{x}_i, y_i, \mathbf{r}_t)}{jj\mathbf{g}(\mathbf{x}_i, y_i, \mathbf{r}_t)jj_2)}\right) \tag{5.4}$$
$$\text{where } \mathbf{g}(\mathbf{x}_i, y_i, \mathbf{r}_t) = r_{\mathbf{r}_t} L(f(\mathbf{x}_i + \mathbf{r}_t; \tilde{\theta}), y_i)$$

Here, α is the step size and t is the step index.

5.4 Experiments and Analysis

5.4.1 Experimental Setup

Tasks and Datasets. For evaluation, we used XNLI [111], the most prominent cross-lingual Natural Language Inference corpus, which extends the MultiNLI dataset [108] to 15 languages. In our experiments, we considered 20k training data, i.e., 5% of the original training size to study lower-resource settings requiring augmentation. Following previous work, we consider English as the source language in our experiments.

Model Details. To show that our reorder augmentation strategy does not require auxiliary data from a low-resource target language, we only give it access to parallel data for another closely related high-resource language. Specifically, we use the English–German bilingual parallel corpus from JW300 [126]. Like English, German commonly adopts an SVO word order, but in some instances also mandates SOV and is generally less rigid than English. This allows us to demonstrate the utility of reorder augmentation even in the absence of data from a language similar to the target language. We relied on *FastAlign*[1] to induce 200k training pairs for Seq2Seq fine-tuning on reordering.

As the pre-trained Seq2Seq model, we used Google's T5-base [123], a unified text-to-text Transformer, to generate new training examples. During generation, we set the beam size as 1 and use sampling instead of greedy decoding. For the Teacher model in semantic augmentation, we relied on RoBERTa-Large [122], a robustly optimized BERT model, to fine-tune NLI on English. As the multilingual model, we employ XLM-RoBERTa-base (XLM-R) [3], trained on over 100 different languages. For PGD, the step size α, norm constraint size ϵ, and number of steps K are 1.0, 3.0, 3, respectively. All hyperparameter tuning is conducted based on the accuracy on the English validation set. The Teacher strategy for XNLI then is used for the rectification of semantically augmented texts, as inference

[1] https://github.com/clab/fast_align

requires particularly clean data. The threshold T for this is 0.8. An overview of the basic network parameter values is given in Table 6.3. We rely on early stopping as a termination criterion. For all NLI classification results, we randomly repeat each experiment 5 times and report the averaged accuracy.

Parameter	RoBERTa	T5	XLM-R
max. sequence length	128	150	128
training batch size	16	8	32
learning rate	1e-5	3e-4	1e-5
max. grad. norm	-	1.0	-

Table 5.1: Hyper-parameters for pretrained models.

Approach	*en*	*de*	*es*	*zh*	*fr*	*ru*	*ar*	*sw*	*ur*	*bg*	*el*	*th*	*tr*	*vi*	*hi*	**avg**
In-Language Supervised Learning (Translate–Train)																
RoBERTa	88.2															
mBERT	73.3	65.2	69.0	66.5	66.5	64.8	61.7	57.7	56.3	65.8	63.4	49.3	61.5	66.9	59.3	63.1
XLM-R	77.7	70.6	73.0	68.1	72.8	70.6	67.4	61.8	60.5	73.2	71.0	68.9	69.3	70.2	64.9	69.3
Zero-Shot Cross-Lingual Transfer																
XLM-R	77.7	71.7	72.6	69.5	72.7	70.2	67.7	60.7	61.0	72.0	70.2	67.4	69.0	71..0	64.9	69.1
+PGD	78.9	71.8	74.5	70.2	73.5	71.1	67.3	60.7	62.0	72.9	71.3	68.7	69.2	71.3	64.9	69.9
+EA(80k)	77.8	70.3	73.1	69.2	72.9	70.3	67.5	61.6	63.5	72.1	70.1	68.1	68.7	69.5	65.1	69.3
+RA(20k)	78.4	71.0	73.1	67.3	73.0	70.2	67.1	61.5	61.1	71.9	70.3	65.5	67.5	69.5	64.7	68.8
+SA(80k)	79.5	72.0	74.4	69.6	74.1	71.9	67.5	63.6	62.7	73.6	71.9	69.0	69.2	71.0	66.1	70.4
+EA+PGD	77.9	71.9	74.4	71.1	73.5	71.5	68.8	63.3	64.4	74.1	68.3	69.5	68.9	70.4	66.9	70.3
+RA+PGD	78.9	72.5	74.7	71.1	74.5	72.0	68.6	63.1	63.6	73.3	72	69.0	69.9	71.7	65.9	70.7
+SA+PGD	80.4	73.4	75.7	71.8	74.0	73.1	69.3	64.5	63.7	74.5	73.2	70.3	70.2	72.3	66.9	71.5
+EA+SA+PGD	80.0	74.0	76.1	73.0	75.5	73.9	**70.2**	63.7	65.5	75.4	73.3	70.5	71.4	72.9	68.0	72.2
+RA+SA+PGD	**80.8**	**74.5**	**77.3**	**73.6**	**75.8**	**74.9**	70.0	**64.8**	**65.7**	**76.3**	**74.9**	**71.6**	**71.4**	**74.5**	**68.5**	**73.0**

Table 5.2: Accuracy (in %) on XNLI with augmented examples used for cross-lingual transfer. The number of augmented examples from EA, RA and SA are 80k, 20k, 80k. EA [25] is Easy Data Augmentation. The best cross-lingual transfer results under XLM-R are given in boldface.

5.4.2 Main Results

Cross-lingual Inference Classification. Table 5.2 compares our approach against several strong baselines on XNLI. The first part considers in-language supervised learning, where we relied on genuine training data from the target language rather than a cross-lingual setting. These results are merely provided for comparison. The second part considers zero-shot cross-lingual transfer, i.e., the setting we are targeting in this chapter: We first used English training data to train the XLM-R model and then applied it to non-English languages without

any training data in the target language. We also trained the model with PGD adversarial training to assess how well PGD works without any data augmentation. Next, we evaluate XLM-R when trained on original and augmented examples from several augmentation methods, with and without adversarial training, respectively. The first of these is Easy Augmentation (EA) by Wei and Zou [25], a state-of-the-art method for data augmentation in NLP. It mixes 4 strategies, namely synonym replacement, random insertion, random swapping, and random deletion, applying each of these to 20% of words in a sentence. Additionally, we consider our proposed RA and SA strategies, as well as combinations of EA or RA with SA.

Compared with vanilla XLM-R without adversarial training, XLM-R with PGD works better across a range of non-English languages, which shows the effectiveness of adversarial training for more robustness in cross-lingual settings. We observe that XLM-R, when trained with EA or RA, outperform the setting without augmentation for English and some non-English languages, though it does not achieve sufficiently stronger results in terms of the average accuracy across different languages. This suggests that XLM-R struggles to benefit from the augmented instances from RA for better generalizability. In contrast, when trained with SA, XLM-R performs better than without SA examples for most languages, confirming that our semantic augmentation is beneficial. Remarkably, XLM-R with SA examples even succeeds at outperforming in-language training with an average absolute improvement of about 1.1% in accuracy, suggesting that cross-lingual models trained with automatically generated English examples can be more informative with regard to inference than target language examples.[2] Next, we also observe that the accuracy of XLM-R with additional examples from EA, RA, SA is boosted with PGD. This suggests that adversarial training is particularly useful to boost generalizability and robustness when operating on artificial augmented examples.

Beyond this, our full zero-shot approach further outperforms all baselines across 14

[2]Note that the in-language training data in XNLI was created using machine translation.

languages, including in-language training. This demonstrates the value of improving generalizability and robustness by adding diverse forms of augmentation in an adversarial training framework that can cope with noisy examples.

Approach	p	en	de	es	zh	fr	ru	ar	sw	ur	bg	el	th	tr	vi	hi	**avg**
Teacher ($T = 0$)	100%	79.7	72.8	75.6	71.7	73.9	73.0	69.3	64.5	63.8	74.0	72.6	69.8	70.0	71.8	66.5	**71.3**
Teacher ($T = 0.8$)	94%	80.4	73.4	75.7	71.8	74.0	73.1	69.3	64.5	63.7	74.5	73.2	70.3	70.2	72.3	66.9	**71.6**
TR ($T = 0.8$)	100%	79.1	72.9	75.3	71.4	74.1	73.1	68.8	64.1	63.6	73.9	73.1	70.4	70.4	72.0	66.6	**71.3**
Agreement	66%	78.7	71.3	74.5	70.8	72.7	71.7	68.7	63.8	62.6	73.0	72.0	69.7	69.4	71.1	65.9	**70.4**
Requested	100%	75.4	67.5	70.1	69.0	68.0	69.2	65.7	61.1	61.6	70.5	68.3	65.9	68.3	70.6	64.1	**67.7**

Table 5.3: Accuracy (in %) on XLNI with different rectifying strategies, training on XLM-R with SA and PGD. T is the threshold. p denotes the percentage of initial augmented examples retained for training.

Approach	pc	en	es	de	zh	fr	ja	ko	**avg**
Teacher ($T = 0$)	100%	94.50	90.05	88.05	82.3	90.0	77.15	76.85	**85.56**
Teacher ($T = 0.8$)	97%	94.50	89.65	88.70	80.85	89.65	77.75	76.75	**85.41**
TR ($T = 0.8$)	100%	94.90	89.55	88.10	82.95	90.10	77.05	77.80	**85.78**
Agreement	27%	94.75	88.75	88.25	82.35	90.00	77.85	78.30	**85.75**
Requested	100%	54.65	54.65	55.25	55.3	54.85	54.85	55.85	**55.06**

Table 5.4: Accuracy (in %) on PAWS-X with different rectifying strategies, training on XLM-R with augmentation and PGD.

5.4.3 Ablation Studies and Analysis

Comparisons on Different Rectifying Strategies. One key part of our method is the label rectification mechanism. We compare different rectification strategies in Table Table 5.3. The results show that the Teacher and TR methods introduced in Section subsection 5.3.2 yield fairly similar results. This confirms the robustness of our approach with regard to the choice of strategy. The same also holds for an additional option, **Agreement**, which retains only those examples on which the prediction from the Teacher agrees with the originally requested label. Finally, for comparison, we evaluated yet another strategy, **Requested**, which always adopts the originally requested labels as chosen for generation. We find that this strategy introduces overly many unreliable labels, so the model is unable to work well. This confirms that rectifying labels with a Teacher model is a crucial ingredient.

Approach	*en*	*de*	*es*	*zh*	*fr*	*ru*	*ar*	*sw*	*ur*	*bg*	*el*	*th*	*tr*	*vi*	*hi*	**avg**
XLM-R (10k)	74.5	68.0	70.3	65.5	70.8	68.0	64.2	61.1	60.2	69.9	68.9	65.0	66.9	68.4	61.5	**66.9**
+SA +FGM	77.5	70.9	73.6	68.3	73.1	70.7	67.3	62.2	62.2	72.8	70.5	68.4	67.3	70.2	64.9	**69.3**
+SA +PGD	78.2	71.4	73.7	70.8	73.1	71.2	68.1	62.3	63.2	73.6	71.8	68.9	69.2	71.1	65.6	**70.1**
+RA +SA +PGD	79.1	73.0	75.2	72.3	73.6	72.5	69.2	64.5	63.7	74.5	72.3	70.0	70.7	72.7	67.2	**71.4**
Improvement(%)	6.2	7.4	7.0	10.0	4.0	6.6	7.8	5.6	5.8	6.6	4.9	7.7	5.7	6.3	9.3	**6.7**
XLM-R (20k)	77.7	70.0	72.5	69.2	72.7	70.6	66.9	61.6	60.8	72.0	70.2	66.7	68.7	70.6	64.9	**69.0**
+SA +FGM	79.3	72.4	74.7	70.6	73.7	71.8	67.6	63.5	63.0	72.9	71.9	68.3	69.3	71.6	66.6	**70.5**
+SA +PGD	80.4	73.4	75.7	71.8	74.0	73.1	69.3	64.5	63.7	74.5	73.2	70.3	70.2	72.3	66.9	**71.6**
+RA +SA +PGD	80.8	74.5	77.3	73.6	75.8	74.9	70.0	64.8	65.7	76.3	74.9	71.6	71.4	74.5	68.5	**73.0**
Improvement(%)	4.0	6.4	6.6	6.4	4.3	6.1	4.6	5.2	8.1	6.0	6.7	7.3	3.9	5.5	5.5	**5.8**

Table 5.5: Accuracy (in %) on XNLI experiments with different amounts of training and augmentation data, and different adversarial training methods.

Approach	*en*	*de*	*es*	*zh*	*fr*	*ru*	*ar*	*sw*	*ur*	*bg*	*el*	*th*	*tr*	*vi*	*hi*	**avg**
XLM-R (20k)	77.7	71.7	72.6	69.5	72.7	70.2	67.7	60.7	61.0	72.0	70.2	67.4	69.0	71..0	64.9	69.1
+EA (20k)	77.4	69.1	71.9	67.5	71.6	69.3	65.5	61.0	61.5	71.1	69.2	67.1	67.1	68.8	63.9	68.1
+EA (80k)	77.8	70.3	73.1	69.2	72.9	70.3	67.5	61.6	63.5	72.1	70.1	68.1	68.7	69.5	65.1	**69.3**
+RA (20k)	78.4	71.0	73.1	67.3	73.0	70.2	67.1	61.5	61.1	71.9	70.3	65.5	67.5	69.5	64.7	**68.8**
+RA (80k)	77.5	70.8	73.3	68.1	72.2	70.3	66.8	60.7	60.3	72.5	70.5	66.0	67.6	69.3	63.3	68.6
+SA (20k)	78.2	70.6	72.8	67.3	72.6	70.3	66.5	61.4	60.4	71.8	69.6	66.9	67.6	69.5	64.0	68.6
+SA (80k)	79.5	72.0	74.4	69.6	74.1	71.9	67.5	63.6	62.7	73.6	71.9	69.0	69.2	71.0	66.1	**70.4**
+PGD	78.9	71.8	74.5	70.2	73.5	71.1	67.3	60.7	62.0	72.9	71.3	68.7	69.2	71.3	64.9	69.9
+EA +PGD (20k)	77.6	70.9	73.9	69.8	73.0	71.1	67.1	62.4	63.8	73.0	71.3	68.9	69.1	71.2	65.8	69.9
+EA +PGD (80k)	77.9	71.9	74.4	71.1	73.5	71.5	68.8	63.3	64.4	74.1	68.3	69.5	68.9	70.4	66.9	**70.3**
+RA +PGD (20k)	78.9	72.5	74.7	71.1	74.5	72.0	68.6	63.1	63.6	73.3	72	69.0	69.9	71.7	65.9	**70.7**
+RA +PGD (80k)	78.4	71.9	74.9	71.0	73.7	71.9	68.7	62.6	64.0	73.4	72.1	68.9	69.9	71.9	66.4	70.4
+SA +PGD (20k)	79.3	73.3	74.0	69.4	73.3	71.0	67.6	62.7	62.4	73.7	71.7	68.3	69.28	71.1	65.6	70.2
+SA +PGD (80k)	80.4	73.4	75.7	71.8	74.0	73.1	69.3	64.5	63.7	74.5	73.2	70.3	70.2	72.3	66.9	**71.5**

Table 5.6: Accuracy (in %) on XNLI experiments trained using 20k vs. 80k augmentation data from EA, RA, SA, with and without PGD.

Comparisons on Adversarial Perturbations. For assessing the value of PGD for adversarial perturbation, Table Table 5.5 compares PGD with the standard Fast Gradient Method (FGM) for adversarial perturbation [113] as introduced in Section subsection 5.3.3. We ran experiments on XNLI with 10k and 20k training data, each augmented with 80k induced semantic examples. We observe that FGM obtains a lower average accuracy than PGD with the same amount of training data, confirming the superiority of PGD in providing better adversarial perturbations than FGM to improve both generalization and robustness.

Effectiveness on Different Training Sizes. Data augmentation is an important approach to deal with scarce labels. The results in Table Table 5.5 further show that when fine-tuning T5 using 10k XNLI training instances with 80k semantic and 10k reorder augmented examples, we obtain substantially better results than when using 20k training instances

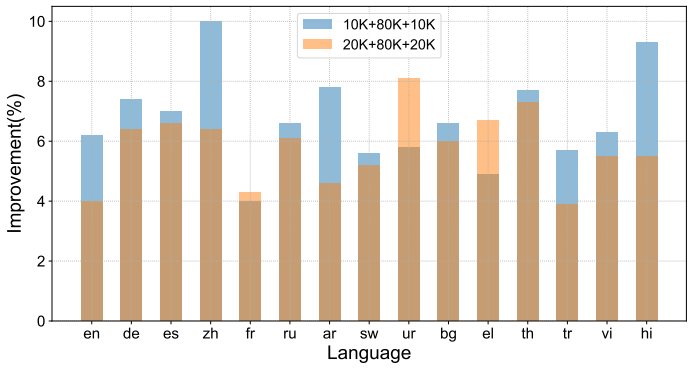

Figure 5.2: Relative improvements of XLM-R with augmentation and PGD over XLM-R. Blue refers to the improvement on 10k original instances plus 80k SA and 10k RA, while orange refers to the improvement on 20k original instances plus 80k SA and 20k RA, and brown designates the overlap between blue and orange.

without augmentation. We can also observe the improvement of XLM-R with RA, SA, and adversarial training over vanilla XLM-R on each language as plotted in Figure 5.2. The relative gains with 10k training data are larger than with 20k training data across a range of languages, which shows that our method is consistently most beneficial when training data is scarce.

Influence of Amount of Augmentation. To assess the role of the amount of data augmentation, we conducted experiments on XNLI with 20k training examples, and evaluated the effect of adding either 20k or 80k augmented examples from EA, RA, SA. The results are given in Table Table 5.6. When trained without PGD, one can often benefit from using up to 80k augmented examples. Due to the inherent reordering differences between English and German, there are limits regarding the amount of such data one ought to incorporate. We find that 20k instances from RA can suffice. We observe that EA with PGD requires up to 80k augmented instances, i.e., 3 times the size of the original training data, to outperform XLM-R with PGD, whereas only 20k augmented examples suffice for RA with PGD to beat XLM-R with PGD.

Case Studies. To better illustrate the principles of our data augmentation technique, we provide several examples. Table Table 5.7 shows two examples of the three data

augmentation processes on XNLI. For the first example, the original label is **contradiction**, so **entailment** and **neutral** serve as requested labels to generate new training text. Next, our Teacher model attempts to rectify these labels. Although our generative model treats *Vrenna and I fought him in a fight, but he had just gotten us* as neutral to S_1 (*Vrenna and I both fought him and he nearly took us*), the Teacher model changes the label to **entailment**. For the second example, both the generative and Teacher model are unable to conclude that *The rice ripens in the summer* is contradictory with the premise. From the two EA outputs, we can observe *him* is randomly deleted in Example (1) and *the* and *rice* is swapped in Example (2), which loses some information, whereas RA Seq2Seq generated examples maintain all crucial information despite the reordering.

	V	RL	L	Text
(1)	O	–	**contradiction**	$\mathbf{S_1}$: Vrenna and I both fought him and he nearly took us. $\mathbf{S_2}$: Neither Vrenna nor myself have ever fought him.
	EA	–	**contradiction**	$\mathbf{S_1}$: Vrenna and I both fought him and took nearly he us. $\mathbf{S_2}$: Neither Vrenna nor myself have ever fought.
	RA	–	**contradiction**	$\mathbf{S_1}$: Vrenna and I both him fought and he us nearly took. $\mathbf{S_2}$: Neither me nor Vrenna have him ever fought.
	SA	**entailment**	**entailment**	$\mathbf{S_2}$: It was the guy that nearly took the couple of us.
	SA	**neutral**	**entailment**	$\mathbf{S_2}$: Vrenna and I fought him in a fight, but he had just gotten us.
(2)	O	–	**contradiction**	$\mathbf{S_1}$: In summer the rice forms a green velvety blanket, then turns golden in autumn when it ripens and is harvested. $\mathbf{S_2}$: The rice is golden and harvestable in the summer, but turns green in autumn.
	EA	–	**contradiction**	$\mathbf{S_1}$: Harvested summer the rice forms a green velvety blanket then turns golden in autumn when is ripens and it in. $\mathbf{S_2}$: The the is golden and harvestable in rice summer, but turns green in autumn.
	RA	–	**contradiction**	$\mathbf{S_1}$: In summer forms the rice a green velvety blanket, turns then in autumn golden when it ripens and harvested is. $\mathbf{S_2}$: The rice is golden and harvestable in the summer, but turns in autumn green.
	SA	**entailment**	**entailment**	$\mathbf{S_2}$: The rice turns golden in autumn when it ripens.
	SA	**neutral**	**entailment**	$\mathbf{S_2}$: The rice ripens in the summer and then turns golden in the autumn.

Table 5.7: Examples of XNLI data augmentation. V: Version (O: Original). RL: Requested Label. L: Final (possibly rectified) label.

5.5 Discussion

While multilingual pretrained model have enabled better cross-lingual learning, we still often encounter data scarcity issues due to the high cost of collecting data, which weakens the generalization ability of the multilingual model.

To address this, this chapter proposes a novel data augmentation strategy with label

rectification to build synthetic examples, outperforming even models trained with larger amounts of ground-truth data. We show that we can best learn from such noisy instances with adversarial training, which enables the classifier to transfer more information from the source language to other languages and to become more robust. Remarkably, with this, our models trained without any target language training data at all are able to outperform models trained fully on in-language training data. Moreover, the amount of augmented data from our Seq2Seq-based reorder augmentation used in training is much less than that required by the state-of-the-art EDA method in order to achieve comparable performance. Finally, in our series of follow-up experiments comparing different training regimens and variants, one notable finding is that our overall augmented approach can even outperform non-augmented supervision with twice as many ground truth labels. Overall, this suggests our combination of data augmentation with adversarial training as a valuable way of learning substantially more accurate and more robust models without any target-language training data.

Research on cross-lingual NLP is often motivated by a desire to provide state-of-the-art advances to linguistic communities that have been underserved. Such advances may enable better access to information as well as to products and services. However, there is a risk that such technological advances may not always be desired by the relevant communities and may indeed also cause harm to them [127]. Moreover, cross-lingual systems in particular may exhibit biases with regard to the source language used for training and the general cultural assumptions reflected in such data. In light of this, special care needs to be taken to analyze potential outcomes and risks before deploying cross-lingual systems in real-world applications.

CHAPTER 6

MULTI-SOURCE AUXILIARY LEARNING IN TEMPORAL EVENT REASONING

6.1 Overview

Auxiliary learning is a common means of improving the performance on a primary task of interest [128, 129, 130]. In our work, we propose two auxiliary tasks to acquire better temporal reasoning abilities: (*i*) part-of-speech (POS) tagging, and (*ii*) question constraints. POS tagging as an auxiliary task is able to ensure a better understanding of tense-related information within a sentence. For example, as shown in Table 6.1, the word "predicted" in "People have predicted his demise so many times ..." is labeled as VBN (past participle), while "passed" is labeled as VBD (past tense) in "Security Council passed a resolution ...". Being able to capture such distinctions enables the model to more accurately distinguish what happened from what *has* (perhaps more recently) happened.

The second auxiliary task, question constraints, can be viewed as a self-supervised task and is induced based on a temporal question answering dataset.

Example	**POS Tag**	**Temporal Information**
People have **predicted** his demise so many times...	VBN: verb, past participle	event has happened
Security Council **passed** a resolution ...	VBD: verb, past tense	event happened

Table 6.1: Excerpts from input passages with different verb POS tags.

Passage: They were traveling in an up-armored high-mobility, multi-purpose, wheeled vehicle when this occurred. Those injured were evacuated by air to a nearby forward operating base for treatment.	
Questions	**Answers**
What events have already finished?	traveling, occurred, evacuated
What will happen in the future?	No answer.
What events happened during their travel?	occurred, evacuated
What events have begun but has not finished?	treatment
What happened after it occurred?	evacuated, treatment
What happened before the injured were treated?	traveling, occurred, evacuated

Table 6.2: Question Answering samples from TORQUE [33].

As shown in Table 6.2, for a given text passage, the dataset provides a set of questions,

and different questions tend to call for different answers. For example, the set of answers to "What events have already finished?" and "What will happen in the future?" should typically be disjoint. Hence, we explore the value of question constraint rules between pairs of questions for a passage. We induce such rules automatically based on their answer overlap, and subsequently enforce them by training the model with the auxiliary classification task of identifying the kind of answer overlap.

We propose a novel multi-source auxiliary learning objective that incorporates the two auxiliary tasks to improve the performance in two temporal event reasoning tasks. Our method achieves a new state-of-the-art performance on the TORQUE [33] dataset (QA setup), improving over previous work by 0.8 F1 points (absolute). Having fine-tuned the model on this QA setup so as to learn complex temporal cues, we further demonstrate the generalizability of our approach by showing that the fine-tuned encoder can then be further fine-tuned to improve the top performance on MATRES [32] (Relation Extraction setup) by 2.3 F1 points. Finally, we show that our approach is particularly performant in a low-resource setting, yielding absolute improvements of up to 19.5%.

6.2 Related Work

Temporal Question Answering. Great strides have been made with new architectures and new self-supervised objectives to improve over vanilla BERT [131]. However, while models such as RoBERTa [122] and AlBERT [132] enable a better understanding of predicates and arguments for conventional QA tasks, our experiments show that they fail to yield substantial gains on temporal QA. Recently, Han et al. [133] presented a temporal-related language model with new self-supervised objectives for improved Temporal QA. In contrast to our approach, this method requires pre-defined event and temporal lexicons.

Temporal Relation Extraction. Compared with temporal QA, temporal relation (TempRel) extraction is widely studied in temporal event reasoning. Many TempRel datasets have been collected, such as TB-Dense [30], RED [31], and MATRES [32], and a variety of models

target this task. For instance, Han *et al.* [134] present a joint event and temporal relation extraction model. Wang *et al.* [135] enforce logical constraints within and across temporal relations via differentiable learning objectives. Zhou *et al.* [136] incorporate probabilistic soft logic regularization and global inference.

Auxiliary Learning. There is a long history of research on multi-task learning [137], e.g., the Multi-Task Deep Neural Network (MT-DNN) seeks to learn representations across diverse natural language understanding tasks [138]. In auxiliary learning, there is a single primary task, and the role of the auxiliary tasks is to improve the performance and generalizability of this primary task. Trinh *et al.* [129] propose a method for better capturing long term dependencies in RNNs with an extra unsupervised auxiliary loss. Xu *et al.* [139] propose multi-task recurrent modular networks for any multi-task recurrent models.

6.3 Method

Following standard practice when training a deep network on multiple tasks [138], our model consists of a shared encoder and several task-specific classifiers on top of it as shown in Figure 6.1. There is one such classifier for the primary task as well as two further ones for our proposed auxiliary tasks. This architecture allows the shared encoder to jointly learn from each of the tasks.

Shared encoder. The encoder is from a pre-trained contextual representation model, denoted as $f_{\text{se}}(\ ;\theta_{\text{se}})$. Given an input text sequence $\mathbf{s}$ consisting of T tokens $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T]$, this encoder infers a contextual hidden representation $\mathbf{h_t}$ 2 R^d d of dimensionality d for each input token $\mathbf{x}_t$.

Primary Task. Our primary task-specific classification module $f_{\text{p}}(\ ;\theta_{\text{p}})$ is responsible for the question answering task. It is applied for fine-tuning on top of the pre-trained model $f_{\text{se}}(\ ;\theta_{\text{se}})$ and consists of a fully-connected layer with softmax activation to map $\mathbf{h_t}$ 2 R^d into R$^{j Y_{\text{p}} j}$. Here, Y_{p} is defined as a set of binary output class labels denoting whether a given

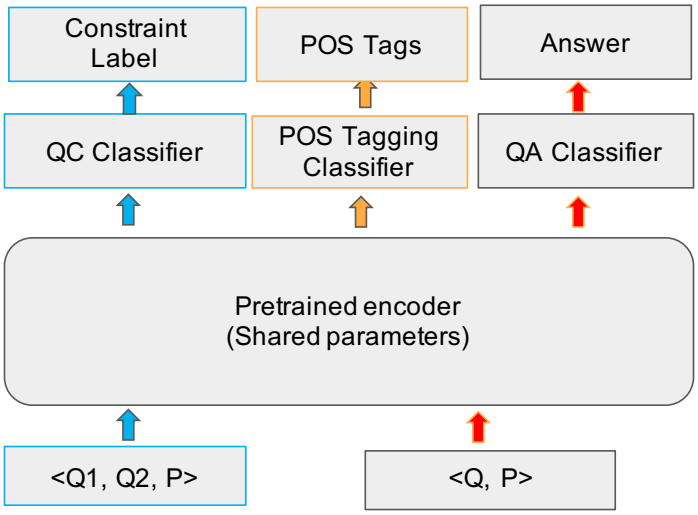

Figure 6.1: Illustration of our auxiliary learning model.

token is deemed a valid answer in response to the question.

Auxiliary Tasks. The model is additionally trained on two auxiliary tasks.

1. *POS tagging.* Our auxiliary POS tagging classification module $f_{\text{pos}}(\ ;\theta_{\text{pos}})$ draws its input from the shared encoder $f_{\text{se}}(\ ;\theta_{\text{se}})$. It then applies a linear mapping $\mathbf{h_t}\ 2\ \mathsf{R}^d$ into $\mathsf{R}^{jY_{\text{pos}}j}$ followed by a softmax activation to predict a distribution over the set of POS tag classes Y_{pos}.

2. *Question Constraint Classification (Question CC).* For a given passage p from our primary QA task, we have a corresponding question set $Q = f(q_i, a_i)\ j\ i\ 2\ f1, ..., ng, a_i\ 6=\ ;g$, where n is the number of questions and a_i is the answer set for question q_i. From this, we can obtain a set of question pairs $C = fhq_i, q_j i\ j\ i < j; i, j\ 2\ f1, ..., nggg$ and a set of answer pairs $A = fha_i, a_j i\ j\ i < j; i, j\ 2\ f1, ..., nggg$. We consider the overlap of answers between two questions to acquire a constraint label for the question pair. In particular, the constraint label is chosen from a set of five relations $Y_{qc} = f$ EQUAL, SUBSET, SUPERSET, DISJOINT, OVERLAPg, based on the corresponding conditions $(a_i = a_j)$, $(a_i \quad a_j), (a_i \quad a_j), (a_i \setminus a_j = ;)$, and $(a_i \setminus a_j\ 6=\ ; ; a_i \setminus a_j\ 6=\ a_i; a_i \setminus a_j\ 6=\ a_j)$. To predict such labels, our model incorporates a question classification module $f_{\text{qc}}(\ ;\theta_{\text{qc}})$ consisting of a fully-connected layer mapping $\mathbf{h_0}\ 2\ \mathsf{R}^d$ into $\mathsf{R}^{jY_{\text{qc}}j}$ with softmax activation.

Auxiliary Learning Objectives. To inject the temporal knowledge into the primary QA training, we jointly learn the primary task along with the two auxiliary tasks. Hence, the overall loss function becomes

$$L = L_p + \lambda_1 L_{\text{pos}} + \lambda_2 L_{\text{qc}}, \tag{6.1}$$

where $L_{\text{p}}, L_{\text{pos}}, L_{\text{qc}}$ are the QA loss, POS tagging loss, and question constraint classification loss, respectively, and λ_1, λ_2 are coefficients to control the influence of each auxiliary task loss term.

6.4 Experiments

6.4.1 Experimental Setup

Tasks and Datasets. For evaluation, we use TORQUE [33], a reading comprehension dataset of temporal ordering questions and answers. It provides 3.2k passages (50 tokens/passage), 24.9k events (7.9 events/passage), and 21.2k user-provided questions. For end-to-end training, the task is modeled as a binary classification problem that requires predicting for each token in the passage whether it is an answer. We also investigate pretraining on TORQUE to then improve on MATRES [32], a temporal relation (TempRel) extraction benchmark, consisting of 275 documents with entity relationships labeled as BEFORE, AFTER, EQUAL, or VAGUE. Regarding metrics, TORQUE is evaluated in terms of F1 score, Exact Match (EM), and Consistency (C). The latter is defined as the percentage of contrast groups for which a model's predictions have F1 80% for all questions in a group. The contrast groups provided by TORQUE consist of questions with contrasting changes to the temporal keywords, e.g., "What happened *after* the snow started?" versus "What happened *before* the snow started?". For MATRES, we report standard micro-averaged F1 scores.

Model Details. For POS tagging as the auxiliary task, we invoke NLTK [140] to obtain POS tags on the TORQUE training set. The size of the POS tag inventory is 36. For

Parameter	TORQUE	MATRES
max. sequence length	180	220
batch size	12	10
learning rate	$1 \quad 10^{5}$	$5 \quad 10^{6}$
# of training epochs	10	5
λ_1	0.001	–
λ_2	0.001	–

Table 6.3: Hyper-parameter settings of our auxiliary learning model.

Method	F1	EM	C
RoBERTa-Large [33]	75.2	51.1	34.5
RoBERTa-Large			
+ Question CC	75.7	**51.3**	36.2
+ POS Tagging	75.8	50.7	35.6
+ POS Tagging + Question CC	**76.0**	51.2	**36.7**

Table 6.4: Results from TORQUE experiments.

question constraint classification, the number of question pairs extracted from the training set for the five labels defined in Section section 6.3 are 4,307, 11,610, 6,181, 42,928, and 7,146, respectively. We adopt RoBERTa-Large [122] as the pre-trained encoder. To further evaluate the effectiveness of auxiliary learning, we use models fine-tuned on TORQUE first to evaluate on MATRES. We tune the hyper-parameters based on the respective development sets and list their values in Table Table 6.3. On TORQUE, as for the original baseline, we report average results over 3 random seeds, while on MATRES, we consider averages over 5 runs.

6.4.2 Results and Analysis

TORQUE (Question Answering Setup). The current SOTA method on TORQUE is RoBERTa-Large [33]. Table Table 6.4 compares our approach against this baseline to evaluate the effectiveness of auxiliary learning. We first evaluate on RoBERTa-Large with either POS tagging or Question CC as the auxiliary task. Compared with RoBERTa-Large, we observe that adding Question CC improves the Consistency score, while POS tagging in particular improves the F1 score. This shows that our answer constraints lead to a

Ratio	30%			50%			100%		
Method	**F1**	**EM**	**C**	**F1**	**EM**	**C**	**F1**	**EM**	**C**
RoBERTa-Large	57.3	37.9	20.1	73.3	46.3	32.0	75.2	51.1	34.5
Our Approach	68.5	39.4	25.1	74.3	48.5	34.5	76.0	51.2	36.7
Improvement (%)	19.5%	4.0%	24.8%	1.4%	4.8%	7.8%	1.1%	0.2%	6.4%

Table 6.5: Results on TORQUE with different ratios of training data.

Method	**F1**
Want et al. [135]	78.8
RoBERTa-Large	80.1
+ TORQUE	80.6
+ TORQUE (Question CC)	80.4
+ TORQUE (POS Tagging)	80.7
+ TORQUE (POS Tagging + Question CC)	**81.1**

Table 6.6: Results on MATRES Dataset.

better understanding of the differences between questions, while the POS tagging auxiliary task enables the model to better capture subtle differences. Our full method outperforms RoBERTa-Large across all three metrics, demonstrating that our multi-source auxiliary learning objective is effective for our primary QA task.

Influence of Amount of Training Data for TORQUE. To assess the effectiveness of our method with limited amounts of training data on TORQUE, we compare our full multi-source auxiliary learning approach with RoBERTa-Large using different ratios of training data. As shown in Table Table 6.5, our method yields significant improvements over RoBERTa-Large in terms of F1 and C scores, especially with 30% of training data, which suggests that our auxiliary tasks are particularly fruitful when training data is scarce, although this also means that less supervision is available for POS tagging and question constraint induction.

MATRES (Relation Extraction Setup). As TORQUE provides more complex temporal information, we assess to what extent we can transfer the knowledge learned on it to the MATRES relation extraction task, so as to evaluate the generalizability of our auxiliary learning. As baselines, in addition to RoBERTa-Large, we consider Wang et al. [135], which incorporates temporal logic constraints among events into the training loss function.

Our model is fine-tuned on TORQUE first and then further fine-tuned on MATRES. This outperforms the baselines, showing that MATRES can benefit from the auxiliary information provided by training on TORQUE first. In this regard, compared to versions with just one additional auxiliary task, our full auxiliary learning model proves the most effective at acquiring an understanding of temporal relationships.

6.5 Discussion

In this chapter, we propose a method to inject additional temporal information with multi-source auxiliary learning objectives into pre-trained models for temporal event reasoning. In particular, we consider part-of-speech prediction and question answer constraint classification as additional objectives, and investigate how pretraining on question answering can benefit temporal relation extraction. Our experiments show that we achieve state-of-the-art results on TORQUE as well as on MATRES, and that our auxiliary learning method is particularly useful in low-resource settings.

CHAPTER 7
AUXILIARY ASYMMETRICAL HIERARCHICAL REVIEW-BASED NETWORKS FOR RATING ESTIMATION

7.1 Overview

In this chapter, we propose an Asymmetrical Hierarchical Network with Attentive Interactions (AHN) for recommendation. AHN progressively aggregates salient sentences to induce review representations, and aggregates pertinent reviews to induce user and item representations. AHN is particularly characterized by its asymmetric attentive modules to flexibly distinguish the learning of user embeddings as opposed to item embeddings. For items, several attention layers are invoked to highlight sentences and reviews that contain rich aspect and sentiment information. For users, we designed an interaction-based co-attentive mechanism to dynamically select a homogeneous subset of contents related to the current target item. In this manner, AHN hierarchically induces embeddings for user–item pairs reflecting the most useful knowledge for personalized recommendation. In summary, our contributions are

1. We identify the asymmetric attention problem for review-based recommendation, which is important but neglected by existing approaches.
2. We propose AHN, a novel deep learning architecture that not only captures both of the asymmetric and hierarchical characteristics of the review data, while also enabling interpretability of the results.
3. We conduct experiments on 10 real datasets. The results demonstrate that AHN consistently outperforms the state-of-the-art methods by a large margin, while providing good interpretations of the predictions.

7.2 Related Work

Exploiting reviews has proven considerably useful in recent work on recommendation. Many methods primarily focus on topic modeling based on the review texts. For example, HFT [141] employs LDA to discover the latent aspects of users and items from reviews. RMR [142] extracts topics from reviews to enhance the user and item embeddings obtained by factorizing the rating matrix. TopicMF [143] jointly factorizes a rating matrix and bag-of-words representations of reviews to infer user and item embeddings. Despite the improvements achieved, these methods only focus on topical cues in reviews, but neglect the rich semantic contents. Moreover, they typically represent reviews as bag-of-words, and thus remain oblivious of the order and contexts of words and sentences in reviews, which are essential for modeling the characteristics of users and items [35].

Inspired by the astonishing advances of recent deep NLP techniques in various applications [144, 145, 146, 147, 2, 148], there has been increasing interest in studying deep learning models. DeepCoNN [35] employs CNNs as an automatic feature extractor to encode each user and item into a low-dimensional vector by assessing the relevant set of historical reviews. TransNet [36] extends DeepCoNN by augmenting the CNN architecture with a multi-task learning scheme to regularize the user and item embeddings towards the target review. These methods, however, lack interpretability [149] in their results.

To better understand the predictions, several attention-based methods have been developed. D-ATT [37] incorporates two kinds of attention mechanisms on the words of reviews to find informative words. NARRE [38] invokes review-level attention weights to aggregate review embeddings to form user (item) embeddings. HUITA [150] is equipped with a symmetric hierarchical structure, where, at each level (e.g., word level), a regular attention mechanism is employed to infer the representation of the subsequent level (e.g., sentence level). MPCN [151] models the interactions between a user's reviews and an item's reviews via co-attention based pointers that are learned with the Gumbel-Softmax trick

[152]. However, all these methods just learn user and item embeddings in parallel and fail to consider the important differences between the two. As discussed before, this leads to suboptimal predictions.

Unlike the aforementioned methods, our method learns several hierarchical aggregators to infer user (item) embeddings. The aggregators are asymmetric to flexibly pay varying levels of attention to a user's (item's) reviews, so as to enhance the prediction accuracy and model interpretability.

7.3 Model

In this section, we introduce our AHN model in a bottom-up manner. Figure 7.1 illustrates the architecture of AHN.

7.3.1 Sentence Encoding

The sentence encoding layer (omitted in Figure 7.1) aims to transform each sentence (in each review) from a sequence of discrete word tokens to a continuous vector embedding. We use a word embedding model to lay the foundation of this layer. Suppose the sentence s has l words. By employing a word embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times |V|}$, s can be represented by a sequence $[\mathbf{e}_1, ..., \mathbf{e}_l]$, where $\mathbf{e}_i$ is the embedding of the i-th word in s, d is the dimensionality of the word embedding, and V is the whole vocabulary of words. The matrix $\mathbf{E}$ can be initialized using word embeddings such as word2vec [118] and GloVe [69], which are widely used in NLP. To refine the word embeddings, $\mathbf{E}$ is fine-tuned during model training.

To learn an embedding for s, we employ a bi-directional LSTM [146] on its constituent word embeddings, and apply max-pooling on the hidden states to preserve the most informative information. That is

$$\mathbf{s} = \max([\tilde{\mathbf{e}}_1, ..., \tilde{\mathbf{e}}_l]), \tag{7.1}$$

where $\mathbf{s}$ is the embedding of s and

$$\tilde{\mathbf{e}}_i = \text{BiLSTM}(\tilde{\mathbf{e}}_{i\ \ 1}, \mathbf{e}_i)\ \ (1 \quad i \quad l), \tag{7.2}$$

where $\tilde{\mathbf{e}}_0$ is initialized by an all-zero vector $\mathbf{0}$.

Suppose a review has k sentences. We can then represent this review by a sequence $[\mathbf{s}_1, ..., \mathbf{s}_k]$, where $\mathbf{s}_i$ is the embedding of the i-th sentence in the review, as inferred by Eq. (Equation 7.1). However, using Eq. (Equation 7.1), each $\mathbf{s}_i$ only encodes its own semantic meaning, but remains oblivious of any contextual cues from its surrounding sentences in the same review. To further refine the sentence embedding, we introduce a context-encoding layer by employing another bi-directional LSTM on top of the previous layer to model the temporal interactions between sentences, i.e.,

$$\tilde{\mathbf{s}}_i = \text{BiLSTM}(\tilde{\mathbf{s}}_{i\ \ 1}, \mathbf{s}_i)\ \ (1 \quad i \quad k), \tag{7.3}$$

where $\tilde{\mathbf{s}}_i$ is the final embedding of the i-th sentence in the review and $\tilde{\mathbf{s}}_0$ is initialized as $\mathbf{0}$.

7.3.2 Sentence-Level Aggregation

Next, we develop sentence-level aggregators to embed each review into a compact vector from its constituent sentences. As discussed before, an ideal method should learn review embeddings in an asymmetric style. Thus, we design AHN to learn different attentive aggregators for users and items, respectively, as highlighted in Figure 7.1.

Sentence Aggregator for Items. Given an item, we are interested in sentences that contain other users' sentiments on different aspects of the item, which are the key factors to determine its overall rating. To build an informative embedding for each review upon such sentences, we use a sentence-level attention network to aggregate the sentence embeddings $[\tilde{\mathbf{s}}_1^v, ..., \tilde{\mathbf{s}}_k^v]$ as follows, where the superscript v is used to distinguish an item's notation from

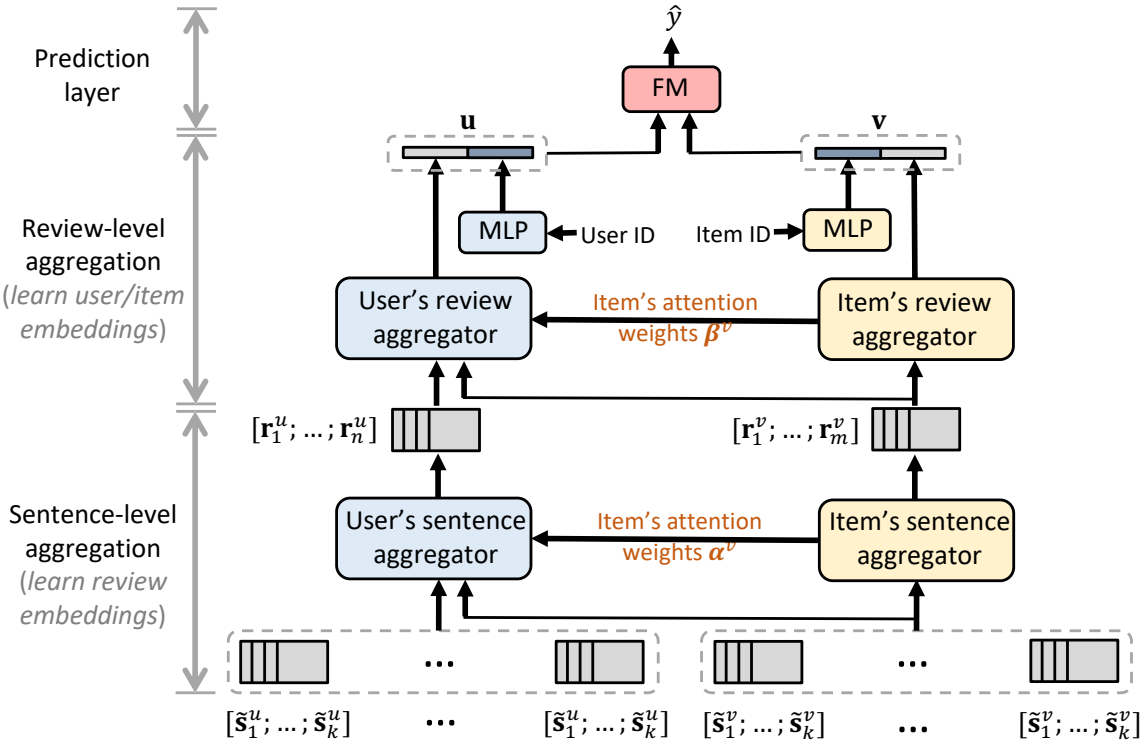

Figure 7.1: The overall architecture of AHN.

a user's notation.

$$\mathbf{r}^v = \sum_{i=1}^{k} \alpha_i^v \tilde{\mathbf{s}}_i^v, \tag{7.4}$$

Here, $\sum_{i=1}^{k} \alpha_i^v = 1$, and α_i^v is the attention weight assigned to sentence $\tilde{\mathbf{s}}_i^v$. It quantifies the informativeness of sentence $\tilde{\mathbf{s}}_i^v$ with respect to v's overall rating, compared to other sentences. The weights α_i^v are computed by our attentive module, taking the sentence embeddings as the input, as

$$\alpha_i^v = \frac{\exp\left(\mathbf{v}_s^\top (\tanh\left(\mathbf{W}_s \tilde{\mathbf{s}}_i^v\right) \quad \sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v))\right)}{\sum_{j=1}^{k} \exp\left(\mathbf{v}_s^\top (\tanh\left(\mathbf{W}_s \tilde{\mathbf{s}}_j^v\right) \quad \sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v))\right)}. \tag{7.5}$$

Here, $\mathbf{v}_s \ 2 \ \mathsf{R}^{h \ 1}$, $\mathbf{W}_s \ 2 \ \mathsf{R}^{h \ d}$, and $\hat{\mathbf{W}}_s \ 2 \ \mathsf{R}^{h \ d}$ are parameters, is the element-wise product, and $\sigma(\ )$ is the sigmoid function. As suggested by Ilse *et al.* [153], the approximate linearity of $\tanh(\ )$ in $[\ \ 1, 1]$ could limit the expressiveness of the model, which can be alleviated by introducing a non-linear gating mechanism. Thus, in Eq. (Equation 7.5), a gate function $\sigma(\hat{\mathbf{W}}_s \tilde{\mathbf{s}}_i^v)$ is incorporated, which is indeed found effective in our experiments.

Sentence Aggregator for Users. Next, we develop an interaction-based sentence aggregator for users. Given a user–item pair, we aim to select a homogeneous subset of sentences from each of the user's reviews such that the selected sentences are relevant to the item to be recommended, i.e., the *target item*. In the following, we introduce a co-attentive network that uses the target item's sentences to guide the search of user's sentences.

After the sentence encoding layer, we can represent each review by a matrix $\mathbf{R} = [\tilde{\mathbf{s}}_1; ...; \tilde{\mathbf{s}}_k]$ *2* $\mathsf{R}^{d \quad k}$, where $[\ ;\ ]$ is the concatenation operation. Suppose a user has n reviews and an item has m reviews. Our method first concatenates all sentences of the item to form $[\mathbf{R}_1^v; ...; \mathbf{R}_m^v]$ *2* $\mathsf{R}^{d \quad mk}$, whose constituent sentences are all relevant to the target item, and thus can be used to guide the search of similar sentences from the user's reviews. To this end, we iterate over each $\mathbf{R}_i^u$ $(1 \quad i \quad n)$ to calculate an affinity matrix as follows, where the superscript u indicates the user notation.

$$\mathbf{G}_i = \phi(f(\mathbf{R}_i^u)^{\top}\mathbf{M}_s f([\mathbf{R}_1^v; ...; \mathbf{R}_m^v])), \quad (1 \quad i \quad n) \tag{7.6}$$

Here, $\mathbf{M}_s$ *2* $\mathsf{R}^{d_s \quad d_s}$ is a learnable parameter, $\phi(\ )$ is an activation function such as ReLU, and $f(\ )$ is a mapping function such as a multi-layer perceptron (MLP). If $f(\ )$ is the identity mapping, Eq. (Equation 7.6) becomes a bilinear mapping. Here, the (p, q)-th entry of $\mathbf{G}_i$ represents the affinity between the p-th sentence of $\mathbf{R}_i^u$ and the q-th sentence of $[\mathbf{R}_1^v; ...; \mathbf{R}_m^v]$.

To measure how relevant the p-th sentence of the user's review $\mathbf{R}_i^u$ is to the target item, we use the maximum value in the p-th row of $\mathbf{G}_i$. The intuition is that, if a user's sentence (i.e., a row of $\mathbf{G}_i$) has a large affinity to at least one sentence of the target item (i.e., a column of $\mathbf{G}_i$) – in other words, the maximal affinity of this row is large – then this user's sentence is relevant to the target item.

However, not all sentences of the target item are useful for searching relevant sentences from the user. For instance, in Figure 2.1, the first sentence of the item's review 2, "*I received it three days ago.*", conveys little information about the target item, and hence

cannot aid in identifying relevant sentences from the user, and indeed may introduce noise into the affinity matrix. To solve this problem, recall that α_i^v in Eq. (Equation 7.5) represents how informative an item's sentence is. Thus, we concatenate α_i^v's of all sentences of the target item to form $\boldsymbol{\alpha}^v \in \mathbb{R}^{1 \times mk}$. Subsequently, we compute an element-wise product between each row of $\mathbf{G}_i$ and the vector $\boldsymbol{\alpha}^v$, i.e., $\mathbf{G}_i \odot_{\text{row}} \boldsymbol{\alpha}^v$. In this manner, the (p, q)-th entry, $(\mathbf{G}_i \odot_{\text{row}} \boldsymbol{\alpha}^v)_{pq}$, is high only if the p-th sentence of the user is similar to the q-th sentence of the target item and the q-th sentence of the target item is non-trivial.

By summarizing the above insights, we learn attention weights for the sentences in $\mathbf{R}_i^u$ for each $i \in [1, n]$ by

$$\boldsymbol{\alpha}_i^u = \text{softmax}(\max_{\text{row}}(\mathbf{G}_i \odot_{\text{row}} \boldsymbol{\alpha}^v)), \tag{7.7}$$

where $\max_{\text{row}}$ refers to row-wise max-pooling for obtaining the maximum affinity. Intuitively, $(\boldsymbol{\alpha}_i^u)_j$ is large if the j-th sentence in the i-th review of the user describes some aspects of some item that is highly similar to the target item. This serves our purpose for selecting a homogeneous subset of sentences from the user.

Next, we use $\boldsymbol{\alpha}_i^u$ to aggregate the sentences in $\mathbf{R}_i^u$ to infer an embedding of the i-th review for the user:

$$\mathbf{r}_i^u = \sum_{j=1}^{k} (\boldsymbol{\alpha}_i^u)_j (\mathbf{R}_i^u)_{:j}, \tag{7.8}$$

where $(\mathbf{R}_i^u)_{:j}$ is the j-th column of $\mathbf{R}_i^u$. Recall that $\mathbf{R}_i^u = [\tilde{\mathbf{s}}_1^u; ...; \tilde{\mathbf{s}}_k^u]$, where each column of $\mathbf{R}_i^u$ is a sentence embedding. Note that our method iterates over i for $i \in [1, n]$ to calculate all review embeddings $\mathbf{r}_1^u$, ..., $\mathbf{r}_n^u$.

Remark. Our co-attentive mechanism employs the idea of sequence pair modeling but notably differs from the conventional co-attention used in QA systems [144, 154, 155]. First, we only consider one side of the affinity matrix, i.e., the user. Second, our affinity matrix is adapted by row-wise multiplication of $\boldsymbol{\alpha}^v$ to quantify the utility of the item's sentences. Thus, our method is designed specifically for learning asymmetric attentions from user–item interactions.

7.3.3 Review-Level Aggregation

From Eq. (Equation 7.4), we obtain review embeddings for an item, $\mathbf{r}_1^v$, ..., $\mathbf{r}_m^v$. From Eq. (Equation 7.8), we obtain review embeddings for a user, $\mathbf{r}_1^u$, ..., $\mathbf{r}_n^u$. As shown in Figure 7.1, based on these review embeddings, we develop review-level aggregators to infer an embedding for each user and item, respectively.

As discussed before, different reviews exhibit different degrees of informativeness in modeling users and items. In particular, an item's reviews are homogeneous. Thus, we are interested in reviews with rich descriptions regarding its relevant aspects and corresponding sentiments, such as the reviews 1–3 of v in Figure 2.1, compared with the less informative review 4 of v. To attend to such reviews, similar to Eq. (Equation 7.4), we aggregate the review embeddings to represent an item by

$$\tilde{\mathbf{v}} = \sum_{i=1}^{m} \beta_i^v \mathbf{r}_i^v, \tag{7.9}$$

where $\sum_{i=1}^{m} \beta_i^v = 1$, and β_i^v is the attention weight assigned to review $\mathbf{r}_i^v$. It quantifies the informativeness of the review $\mathbf{r}_i^v$ with respect to v's overall rating. β_i^v is produced by an attentive module with gating mechanism as follows:

$$\beta_i^v = \frac{\exp\left(\mathbf{v}_r^{\top}(\tanh\left(\mathbf{W}_r\mathbf{r}_i^v\right) \quad \sigma(\hat{\mathbf{W}}_r\mathbf{r}_i^v))\right)}{\sum_{j=1}^{k}\exp\left(\mathbf{v}_r^{\top}(\tanh\left(\mathbf{W}_r\mathbf{r}_j^v\right) \quad \sigma(\hat{\mathbf{W}}_r\mathbf{r}_i^v))\right)}, \tag{7.10}$$

where $\mathbf{v}_r$ 2 $\mathsf{R}^{h\ \ 1}$, $\mathbf{W}_r$ 2 $\mathsf{R}^{h\ \ d}$, and $\hat{\mathbf{W}}_r$ 2 $\mathsf{R}^{h\ \ d}$ are model parameters.

At the same time, a user's reviews are heterogeneous concerning a variety of items that the user has purchased, and not all reviews are relevant to the target item. Thus, similar to Eq. (Equation 7.6) and Eq. (Equation 7.7), given a user–item pair, a review-level co-attentive network is designed to select reviews from the user as guided by the reviews of the item.

Specifically, an affinity matrix at the review level

$$\mathbf{G} = \phi(f([\mathbf{r}_1^u; ...; \mathbf{r}_n^u])^{\top} \mathbf{M}_r f([\mathbf{r}_1^v; ...; \mathbf{r}_m^v])), \tag{7.11}$$

is computed, where $\mathbf{M}_r$ 2 $\mathsf{R}^{d_r \quad d_r}$ is a learnable parameter. Here, the (p, q)-th entry of $\mathbf{G}$ represents the affinity between the p-th review of the user and the q-th review of the item.

Then, attention weights for the reviews of the user

$$\boldsymbol{\beta}^u = \text{softmax}(\max_{\text{row}}(\mathbf{G} \quad _{\text{row}} \boldsymbol{\beta}^v)), \tag{7.12}$$

are obtained, where $\boldsymbol{\beta}^v = [\beta_1^v, ..., \beta_m^v]$ was obtained by Eq. (Equation 7.10) for the item. It is introduced to adapt $\mathbf{G}$ to encode important reviews of the item. Finally, we aggregate the review embeddings to represent a user by the following weighted sum.

$$\tilde{\mathbf{u}} = \sum_{i=1}^{n} \beta_i^u \mathbf{r}_i^u \tag{7.13}$$

Encoding Latent Rating Patterns. Although the embeddings $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ contain rich semantic information from reviews, there are some latent characteristics of users (items) that are not encoded by their reviews, but can be inferred from the rating patterns. For instance, a picky user might tend to uniformly pick lower ratings than a more easygoing user. To encode such personalized preferences, as inspired by Koren *et al.* [34], we embed a one-hot representation of the ID of each user (item) using an MLP, and obtain an embedding vector $\hat{\mathbf{u}}$ ($\hat{\mathbf{v}}$) for the user (item). This vector directly correlates with the ratings of a user (item), and is thus able to capture the latent rating patterns. Then, as illustrated in Figure 7.1, we concatenate $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$ to obtain the final embedding of a user, i.e., $\mathbf{u} = [\tilde{\mathbf{u}}; \hat{\mathbf{u}}]$, and concatenate $\tilde{\mathbf{v}}$ and $\hat{\mathbf{v}}$ to obtain the final embedding of an item, i.e., $\mathbf{v} = [\tilde{\mathbf{v}}; \hat{\mathbf{v}}]$.

7.3.4 Prediction Layer

As shown by the top part of Figure 7.1, the prediction layer receives $\mathbf{u}$ and $\mathbf{v}$, and concatenates them to $[\mathbf{u}; \mathbf{v}]$, which is then fed into a function $g(\ )$ to predict the rating. In this work, we realize $g(\ )$ as a parameterized factorization machine (FM) [39], which is effective to model the pairwise interactions between the input features for improving recommendation performance. Given an input $\mathbf{x}$ 2 $\mathsf{R}^{d\ 1}$, $g(\ )$ is defined as

$$g(\mathbf{x}) = b + \sum_{i=1}^{d} \mathbf{w}_i \mathbf{x}_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \mathbf{z}_i, \mathbf{z}_j \mathbf{x}_i \mathbf{x}_j, \tag{7.14}$$

where b is a bias term, $\mathbf{w}$ is a parameter for linear regression, $\mathsf{f}\mathbf{z}_i\mathsf{g}_{i=1}^{d}$ are the factorized parameter for modeling the pairwise interactions between $\mathbf{x}_i$ and $\mathbf{x}_j$, , denotes the inner product, and the output of $g(\mathbf{x})$ is the predicted rating.

To learn model parameters, we minimize the difference between the true ratings and the predicted ratings, as measured by the mean squared error

$$\ell = \frac{1}{c} \sum_{i=1}^{c} (y_i \quad g([\mathbf{u}; \mathbf{v}]))^2, \tag{7.15}$$

where c is the total number of user–item pairs in the training data, and y_i is the true rating of the i-th user–item pair. The ℓ in Eq. (Equation 7.15) serves as our loss function for model training.

7.4 Experiments

In this section, we evaluate our AHN model on several real datasets and compare it with state-of-the-art approaches.

Dataset	#Users	#Items	#Reviews
Digital_Music (DM)	5,541	3,568	64,706
Office_Products (OP)	4,905	2,420	53,258
Health (HE)	38,609	18,534	346,355
Toys_and_Games (TG)	19,412	11,924	167,597
Kindle_Store (KS)	68,223	61,935	982,619
Pets_Supplies (PS)	19,856	8,510	157,836
Tools_and_Home (TH)	16,638	10,217	134,476
Videos_Games (VG)	24,303	10,672	231,780
Automotive (AM)	2,928	1,835	20,473
Yelp	88,370	33,902	1,332,447

Table 7.1: Statistics of recommendation datasets

Dataset	FM	PMF	NMF	SVD	DeepCoNN	D-ATT	MPCN	HUITA	AHN
Digital_Music (DM)	0.8498	0.8788	1.0491	1.0843	0.8754	0.8506	0.8396	0.8719	**0.8172**
Office_Products (OP)	0.7291	0.7807	0.9285	0.7906	0.7253	0.7124	0.7084	0.7082	**0.6825**
Health (HE)	1.1825	1.2076	1.4317	1.1508	1.0862	1.0915	1.0817	1.1207	**1.0743**
Toys_and_Games (TG)	0.8639	0.9192	1.1105	0.9188	0.8391	0.8364	0.8452	0.8969	**0.8220**
Kindle_Store (KS)	0.6469	0.6695	0.8032	0.7370	0.6514	0.6382	0.6577	0.6544	**0.6270**
Pets_Supplies (PS)	1.3303	1.434	1.6806	1.3191	1.2598	1.2730	1.2566	1.3038	**1.2515**
Tools_and_Home (TH)	1.0229	1.1182	1.3580	1.0373	0.9856	0.9850	0.9871	1.0189	**0.9671**
Videos_Games (VG)	1.1849	1.2473	1.4357	1.3168	1.1575	1.1448	1.1747	1.1772	**1.1138**
Automotive (AM)	0.8189	0.9187	1.2074	0.8140	0.7809	0.7654	0.7643	0.7766	**0.7314**
Yelp	1.6094	1.8207	1.8389	1.6615	1.5957	1.5959	1.6195	1.6105	**1.5735**

Table 7.2: MSE results of the compared methods on different 5-core datasets

Dataset	FM	PMF	NMF	SVD	DeepCoNN	D-ATT	MPCN	HUITA	AHN
Digital_Music (DM)	0.8611	0.8641	0.9491	0.8503	0.8734	0.8429	0.8629	0.8512	**0.7880**
Office_Products (OP)	0.6291	0.6695	0.7346	0.6757	0.6016	0.5914	0.6120	0.6009	**0.5717**
Health (HE)	0.8166	0.9158	0.9200	0.8275	0.8328	0.8019	0.8020	0.8177	**0.7802**
Toy_and_Games (TG)	0.6904	0.6233	0.7575	0.6331	0.6331	0.6292	0.6412	0.6303	**0.5964**
Kindle_Store (KS)	0.5954	0.6035	0.6305	0.6483	0.5325	0.5275	0.5124	0.5312	**0.5092**
Pet_Supplies (PS)	1.2236	1.5239	1.2536	0.9950	0.9927	0.9616	1.0722	1.0168	**0.9421**
Tools_and_Home (TH)	0.8746	0.7668	0.9032	0.7391	0.6632	0.6297	0.6507	0.6445	**0.5948**
Videos_Games (VG)	1.0611	1.0718	1.2435	1.0318	1.0743	1.0365	1.0730	1.0697	**0.9927**
Yelp	1.5432	1.4734	1.5735	1.4025	1.3961	1.4018	1.4033	1.4040	**1.3671**

Table 7.3: MSE results of the compared methods on different 10-core datasets

7.4.1 Datasets

We conducted experiments on 10 different datasets, including 9 Amazon product review datasets for 9 different domains, and the large-scale Yelp challenge dataset[1] on restaurant reviews. Table Table 7.1 summarizes the domains and statistics for these datasets. Across all datasets, we follow the existing work [37, 151] to perform preprocessing to ensure they are

[1]https://www.yelp.com/dataset/challenge

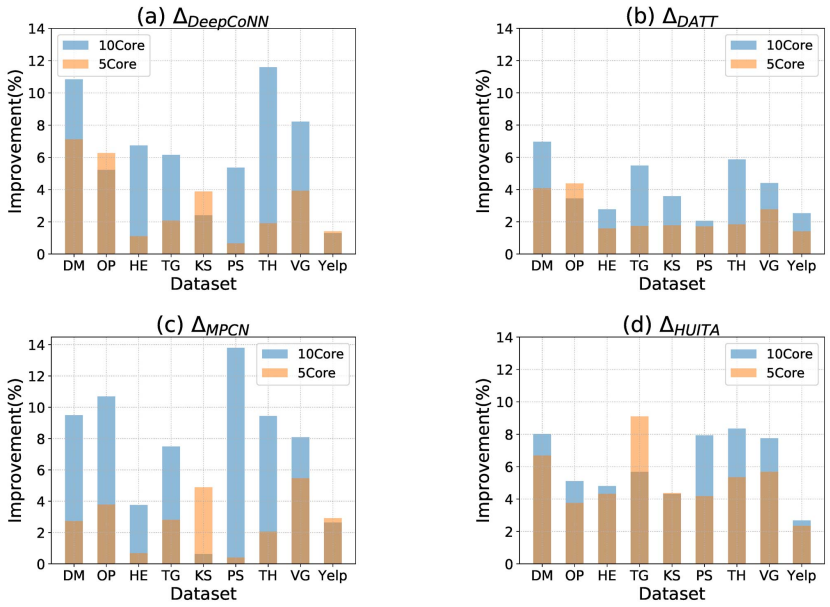

Figure 7.2: The relative improvements of AHN over (a) DeepCoNN, (b) D-ATT, (c) MPCN, and (d) HUITA, on different datasets. The abbreviations of the datasets can be found in Table Table 7.1-Table 7.3. Here, blue refers to the improvement on the 10-core datasets, orange refers to the improvement on the 5-core datasets, brown is the overlapped area between blue and orange.

in a t-core fashion, i.e., the datasets only include users and items that have at least t reviews. In our experiments, we evaluate the two cases of $t = 5$ and $t = 10$. For the Yelp dataset, we follow Seo *et al.* [37] to focus on restaurants in the AZ metropolitan area. For each dataset, we randomly split the user–item pairs into 80% training set, 10% validation set, and 10% testing set. When learning the representations for users and items, we only use their reviews from the training set, and none from the validation and testing sets. This ensures a practical scenario where we cannot include any future reviews into a user's (item's) history for model training.

7.4.2 Compared Methods

We compare our model with both conventional approaches and state-of-the-art approaches, including Factorization Machines (FM) [39], SVD [34], Probabilistic Matrix Factorization (PMF) [156], Nonnegative Matrix Factorization (NMF) [157], DeepCoNN [35], D-ATT

[37], MPCN [151], and HUITA [150].

Among these methods, FM, SVD, PMF, and NMF are rating-based collaborative filtering methods. DeepCoNN, D-ATT, MPCN, and HUITA are state-of-the-art methods that leverage the semantic information in reviews for improved performance. Specifically, DeepCoNN uses the same CNN module to learn user and item embeddings based on their reviews for recommendation. D-ATT extends DeepCoNN by adding a dual-attention layer at word-level before convolution. MPCN attends to informative reviews by several pointers. HUITA uses a symmetric hierarchical structure to infer user (item) embeddings using regular attention mechanisms. It is worth noting that all of the above review-based methods regard user reviews and item reviews as the same type of documents and process them in an identical way.

Finally, to gain further insights on some of the design choices of our AHN model, we compare AHN with its variants, which will be discussed later in the ablation analysis.

7.4.3 Experimental Settings

The parameters of the compared methods are selected based on their performance on the validation set. Specifically, for FM, the dimensionality of the factorized parameters is 10. For SVD, PMF, and NMF, the number of factors is set to 50. DeepCoNN uses 100 convolutional kernels with window size 3. D-ATT uses 200 filters and window size 5 for local attention; 100 filters and window sizes [2, 3, 4] for global attention. MPCN uses 3 pointers, and hidden dimensionality of 300 for inferring affinity matrix. HUITA uses 200 filters in the word-level CNN with window size 3, and 100 filters in the sentence-level CNN with window size 3.

For our AHN model, the dimensionality of the hidden states of the BiLSTM is set to 150. The dimensionality of the user and item ID embeddings are set to 300. The dimensionality of $\mathbf{M}_s$ ($\mathbf{M}_r$) in Eq. (Equation 7.6) (Eq. (Equation 7.11)) is 300. We apply dropout [158] with rate 0.5 after the fully connected layer to alleviate the overfitting problem. The loss

function is optimized by Adam [73], with a learning rate of 0.0002 and a maximum of 10 epochs.

For the methods DeepCoNN, D-ATT, and HUITA, the pre-trained GloVe [69] are used to initialize the word embeddings. For MPCN and our AHN, the word embeddings are learned from scratch since using pre-trained embeddings generally degrades their performance. For all methods, the dimensionality of the word embedding is set to 300. We independently repeat each experiment 5 times, and use the averaged mean square error (MSE) [35] to quantitatively evaluate the performance.

7.4.4 Experimental Results

Table Table 7.2 summarizes the results of the compared approaches on the 5-core datasets. We have several observations from the results. First, review-based methods generally outperform rating-based methods. This validates the usefulness of reviews in providing fine-grained information for refining user and item embeddings for improving the accuracy of rating prediction. Second, methods that distinguish reviews, such as D-ATT and MPCN, often outperform DeepCoNN, which suggests that different reviews exhibit different degrees of importance for modeling users and items. We also observe that HUITA does not show superiority over DeepCoNN. This may stem from its symmetric style of attention learning, which does not make much sense when reviews are heterogeneous. Finally, the proposed AHN consistently outperforms other methods, which demonstrates the effectiveness of distinguishing the learning of user and item embeddings via asymmetric attentive modules so as to infer more reasonable attention weights for recommendation.

Table Table 7.3 presents the results on the 10-core datasets, from which the *Automotive* dataset is excluded because only very few users and items are left after applying the 10-core criterion on it. In contrast to Table Table 7.2, all methods in general achieve better results in Table Table 7.3, since more ratings and reviews become available for each user and item. In this case, we observe that D-ATT often outperforms MPCN. This may be because the

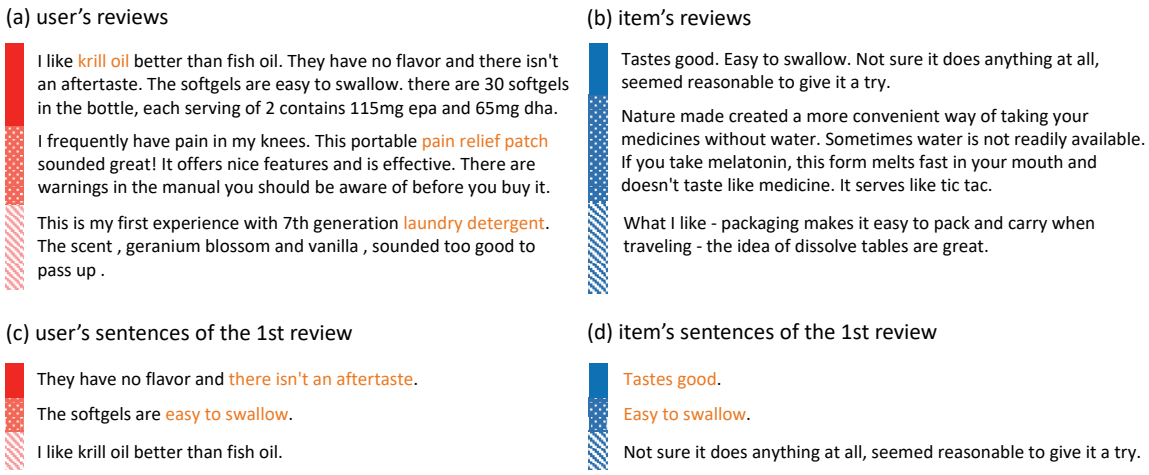

Figure 7.3: The visualization of attention weights on (a) user's reviews, (b) item's reviews, (c) user's sentences, (d) item's sentences. The item is a sleep aid medicine. The vertical bars represent weights. Darker colors indicate higher weights.

Gumbel-Softmax pointers in MPCN make hard selections on reviews, thereby filtering out many reviews that may result in a significant loss of information. This problem is more severe when users (items) have more useful reviews, as in the 10-core scenario. Additionally, we observe that the performance gaps between AHN and the compared methods become larger. Specifically, summarizing the relative improvements of AHN over each of the review-based methods in Figure 7.2, we observe that AHN generally gains more on the 10-core datasets, with absolute gains of up to 11.6% (DeepCoNN), 7.0% (D-ATT), 13.8% (MPCN), and 8.4% (HUITA). This suggests that the more reviews each user and item has, the more important it is to perform proper attention learning on relevant reviews and sentences on both the user and item sides.

7.4.5 Case Study

We next investigate the interpretability of AHN. Figure 7.3(a) and (b) show the attention weights of AHN on the top three reviews of a pair of user and item on the *Health* dataset, where the item is a sleep aid medicine. In each of the user's reviews, the highlighted words indicate the item described by the review. As can be seen, the first two items "krill oil" and "pain relief patch" are more relevant to the item "sleep aid medicine" than the "laundry

Model	VG	DM	AM	OP
AHN	**1.1138**	**0.8172**	**0.7314**	**0.6825**
(a) –Item aggregators	1.1286	0.8205	0.7506	0.6951
(b) –User aggregators	1.1604	0.8246	0.7467	0.6941
(c) –Adapted affinity	1.1363	0.8229	0.7348	0.6936
(d) –FM	1.1267	0.8341	0.7723	0.7078
(e) –Gating	1.1220	0.8188	0.7385	0.6883

Table 7.4: Ablation analysis for AHN

detergent" in the lowest-weighted review. On the other hand, the top two reviews of the item are more informative with regard to the aspects of the item than the last review, which only discusses packaging, a rather marginal aspect of medication. Thus, the review-level attention weights of AHN are meaningful.

Figure 7.3(c) and (d) zoom into the attention weights of AHN on the top three sentences of the first review of the user and item, respectively. The highlighted words indicate the reason of why the sentences are ranked highly. Apparently, the user cares about the taste of the medicine and prefers easily-swallowed softgels, while the item indeed appears to taste good and is easy to swallow. Although the first two sentences in Figure 7.3(d) are short, they convey more useful information than the lowest-weighted sentence. Thus, the sentence-level attention weights are also meaningful. This explains why AHN predicts a 4.4 rating score on this user–item pair, close to the true rating 5.0 given by the user.

7.4.6 Ablation Analysis

Table Table 7.4 presents the results of our ablation analysis using four datasets. In the table, AHN is our original model. In (a), the item's attention modules are replaced by average-pooling. In (b), the user co-attention modules are replaced by attention modules similar to the item ones and this thus constitutes a symmetric model. In (c), we remove the row-wise multiplication between the affinity matrix and the attention weights in Eqs. (Equation 7.7) and (Equation 7.12). In (d), the parameterized factorization machine is replaced by a dot product. In (e), the gating mechanisms in Eqs. (Equation 7.5) and (Equation 7.10) are

removed.

From Table Table 7.4, we observe that different variants of AHN show suboptimal results to various degrees. Comparing with (a), we can observe the importance of considering attention weights on the sentences and reviews of each item. The degraded MSEs of (b) suggest that our asymmetric design in the model architecture is essential. The results of (c) validate our design of the attention-adapted affinity matrix in Eqs. (Equation 7.7) and (Equation 7.12). The substantial MSE drops for (d) establish the superiority of using FM as the prediction layer. The comparison between (e) and AHN suggests the effectiveness of the gating mechanisms. Thus, the results of the ablation study validate the design choices of our model architecture.

7.5 Discussion

In this chapter, we highlight the asymmetric attention problem for review-based recommendation, which has been ignored by existing approaches. To address it, we propose a flexible neural architecture, AHN, which is characterized by its asymmetric attentive modules for distinguishing the learning of user embeddings and item embeddings from reviews, as well as by its hierarchical paradigm to extract fine-grained signals from sentences and reviews. Extensive experimental results on datasets from different domains demonstrate the effectiveness and interpretability of our method.

CHAPTER 8
CONCLUSIONS

This dissertation presents a set of methods involved in leveraging auxiliary signals for low-resource NLP. The goal behind all these methods is to enhance different tasks' performance with additional data resource. Based on this, it introduces a method on how to build domain-specific sentiment embeddings and then incorporate them into the dedicated neural networks for multilingual sentiment analysis, a self-learning framework in cross-lingual text classification with a large amount of unlabeled data considered, a novel data augmentation scheme with adversarial training for cross-lingual NLI, a mutli-source auxiliary learning in temporal event reasoning and auxiliary asymmetrical hierarchical review-based networks for rating estimation. The following are the main contributions and findings of Chapter 3, 4, 5, 6, 7.

In Chapter 3, we have investigated the use of cross-lingually induced sentiment representations to boost the effectiveness of deep neural models for sentiment analysis, incorporated into the network via a separate memory network. Extensive experiments on 9 different languages confirm the effectiveness of this approach, leading to substantial gains across a series of datasets from heterogeneous domains. Our approach has allowed us to generate sentiment embeddings for over 50 languages.

In Chapter 4, we have proposed a self-learning framework to incorporate the predictions of the multilingual BERT model [2] on non-English data into an English training procedure [104]. The initial multilingual BERT model was simultaneously pretrained on 104 languages, and has shown to perform well for cross-lingual transfer of natural language tasks. Our model begins by learning just from available English samples, but then makes predictions on unlabeled non-English samples and a part of those samples with high confidence prediction scores are repurposed to serve as labeled examples for a next iteration of fine-tuning until

the model converges. We also add extra adversarial perturbations into the iteration training process to keep our method more robust.

Based on this multilingual self-learning technique, we demonstrate the superiority of our framework on Multilingual Reuters Financial News Classification and Multilingual Intent Classification in comparison with several strong baselines. Our study then proceeds to show that our method is better on Chinese sentiment classification than other cross-lingual methods that also consider unlabeled non-English data. This shows that our method is more effective at cross-lingual transfer for domain-specific tasks, using a mix of labeled and unlabeled data via a multilingual BERT sentence model.

While multilingual pretrained model have enabled better cross-lingual learning, we still often encounter data scarcity issues due to the high cost of collecting data, which weakens the generalization ability of the multilingual model.

To address this, in Chapter 5, we propose a novel data augmentation method with label rectification to build synthetic examples, outperforming even models trained with larger amounts of ground-truth data. We show that we can best learn from such noisy instances with adversarial training, which enables the multilingual classifier to transfer more information from the source language to other languages and to become more robust. Remarkably, with this, our models trained without any target language training data at all are able to outperform models trained fully on in-language training data. We achieve new state-of-the-art results on cross-lingual NLI classification tasks.

In Chapter 6, 7, we introduce better methods to make use of additional data from different sources in temporal event reasoning and recommendation system. Multi-source auxiliary learning is proposed in Chapter 6. Our method injects additional temporal knowledge into the pre-trained model from two sources. Our approach achieves the new state-of-the-art results on two temporal event QA and classification tasks. In Chapter 7, we present an asymmetrical hierarchical networks with attentive interactions to better exploit and interpret the review information from user's and item's side for review-based rating estimation in

recommendation system. We demonstrate the effectiveness of our method on a variety of real datasets.

8.1 Future Work

There are still some related problems/directions that would be interesting to researchers in studying the possible methods to better leverage auxiliary signals for low-resource NLP.

Semi-Supervised Method. In this dissertation, only a simple self-learning method is considered to make use of unlabeled data, so exploring how to combine new semi-supervised methods with pretrained models is able to be a potential work. Also, we can test the whole semi-supervised framework on more cross-lingual work, such as relation extraction, entity identification detection, etc. Another problem is that the effectiveness of semi-supervised methods crucially depend on the reliability of pseudo-labels of the chosen unlabeled samples, although recently semi-supervised methods make it possible to include completely different unlabeled language samples in the fine-tuning process of pretrained multilingual representation models with labeled data from one language. How to improve the reliability of pseudo-labels is another interesting direction.

Beyond the existing NLP tasks explored. We demonstrate the effectiveness of our methods on a series of NLP applications and our methods also extend to the recommendation system. Considering the flexibility of our proposed methods, we are able to expand them to more work. For example, the scenarios where our self-learning framework can be applied to are not limited in cross-lingual tasks but also in cross-domain problems, such as, some texts from financial domains are different from that in hotel or restaurants domains if we attempt to work on sentiment mining. Besides, our data augmentation with adversarial training work can extend to multimodal settings, which allows us to build up a unified framework on both visual and linguistic modules.

ACKNOWLEDGMENT OF PREVIOUS PUBLICATIONS

P1 X. Dong and G. de Melo, "Cross-lingual propagation for deep sentiment analysis," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, AAAI Press, 2018, pp. 5771–5778

P2 X. Dong and G. De Melo, "A helping hand: Transfer learning for deep sentiment analysis," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 2524–2534

P3 X. Dong and G. de Melo, "A robust self-learning framework for cross-lingual text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp. 6306–6310

P4 X. Dong *et al.*, "Leveraging adversarial training in self-learning for cross-lingual text classification," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, 2020, pp. 1541–1544

P5 X. Dong *et al.*, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 7667–7674

P6 X. Dong *et al.*, "Data augmentation with adversarial training for cross-lingual nli," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 5158–5167

P7 X. Dong *et al.*, "Temporal event reasoning using multi-source auxiliary learning objectives," in *European Conference on Information Retrieval*, Springer, 2022, pp. 102–110

REFERENCES

[1] G. de Melo and S. Siersdorfer, "Multilingual text classification using ontologies," in *Proceedings of ECIR 2007*, (Apr. 2, 2007), Roma, Italy: Springer, 2007, ISBN: 978-3-540-71494-1.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[4] M. Artetxe and H. Schwenk, "Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond," *arXiv preprint arXiv:1812.10464*, 2018.

[5] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *arXiv preprint arXiv:1901.07291*, 2019.

[6] H. Schwenk and X. Li, "A corpus for multilingual document classification in eight languages," *arXiv preprint arXiv:1805.09821*, 2018.

[7] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," *arXiv preprint arXiv:1810.13327*, 2018.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[9] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[10] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[11] C. N. dos Santos and M. A. de C. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *COLING 2014*, 2014.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS 2013*, Lake Tahoe, Nevada, 2013, pp. 3111–3119.

[13] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," in *Proc. EACL 2014*, 2014, pp. 462–471.

[14] A. Klementiev, I. Titov, and B. Bhattarai, "Inducing crosslingual distributed representations of words," in *Proc. COLING 2012*, 2012.

[15] M.-T. Luong, H. Pham, and C. Manning, "Bilingual word representations with monolingual quality in mind," in *Proc. NAACL Workshop on Vector Space Modeling for NLP*, 2015.

[16] G. de Melo, "Wiktionary-based word embeddings," in *Proceedings of MT Summit XV*, (Oct. 30–Nov. 3, 2015), Miami, FL, USA, 2015.

[17] G. de Melo, "Inducing conceptual embedding spaces from Wikipedia," in *Proceedings of WWW 2017*, (Apr. 3, 2017), Perth, Australia: ACM, 2017.

[18] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. ICWSM-14*, 2014.

[19] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 595–605.

[20] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[22] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[23] A. W. Yu, D. Dohan, Q. Le, T. Luong, R. Zhao, and K. Chen, "Fast and accurate reading comprehension by combining self-attention and convolution," in *International Conference on Learning Representations*, 2018.

[24] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 6256–6268.

[25] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of EMNLP-IJCNLP 2019*, Hong Kong, China, Nov. 2019, pp. 6382–6388.

[26] M. Fadaee, A. Bisazza, and C. Monz, "Data augmentation for low-resource neural machine translation," *arXiv preprint arXiv:1705.00440*, 2017.

[27] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," *arXiv preprint arXiv:1805.06201*, 2018.

[28] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional bert contextual augmentation," in *International Conference on Computational Science*, Springer, 2019, pp. 84–95.

[29] A. Anaby-Tavor *et al.*, "Do not have enough data? deep learning to the rescue!" In *AAAI*, 2020, pp. 7383–7390.

[30] N. Chambers, T. Cassidy, B. McDowell, and S. Bethard, "Dense event ordering with a multi-pass architecture," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 273–284, 2014.

[31] T. O'Gorman, K. Wright-Bettner, and M. Palmer, "Richer event description: Integrating event coreference with temporal, causal and bridging annotation," in *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, 2016, pp. 47–56.

[32] Q. Ning, H. Wu, and D. Roth, "A multi-axis annotation scheme for event temporal relations," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1318–1328.

[33] Q. Ning, H. Wu, R. Han, N. Peng, M. Gardner, and D. Roth, "TORQUE: A reading comprehension dataset of temporal ordering questions," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1158–1172.

[34] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, pp. 30–37, 2009.

[35] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, ACM, 2017, pp. 425–434.

[36] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, ACM, 2017, pp. 288–296.

[37] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, ACM, 2017, pp. 297–305.

[38] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the World Wide Web Conference (WWW)*, International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[39] S. Rendle, "Factorization machines," in *International Conference on Data Mining (ICDM)*, IEEE, 2010, pp. 995–1000.

[40] L. Wang, K. Liu, Z. Cao, J. Zhao, and G. de Melo, "Sentiment-aspect extraction based on Restricted Boltzmann Machines," in *Proceedings of ACL 2015*, (Jul. 26, 2015), Beijing, China, 2015, pp. 616–625.

[41] C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan, "Multilingual subjectivity analysis using machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP 2008, Honolulu, Hawaii, 2008, pp. 127–135.

[42] E. Demirtas and M. Pechenizkiy, "Cross-lingual polarity detection with machine translation," in *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, ser. WISDOM '13, Chicago, Illinois: ACM, 2013, 9:1–9:8, ISBN: 978-1-4503-2332-1.

[43] K. Duh, A. Fujino, and M. Nagata, "Is machine translation ripe for cross-lingual sentiment classification?" In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11, Portland, Oregon: Association for Computational Linguistics, 2011, pp. 429–433, ISBN: 978-1-932432-88-6.

[44] M. Haas and Y. Versley, "Subsentential sentiment on a shoestring: A crosslingual analysis of compositional classification," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 694–704.

[45] D. Vilares, C. Gómez-Rodrıguez, and M. A. Alonso, "Universal, unsupervised (rule-based), uncovered sentiment analysis," *Knowledge-Based Systems*, vol. 118, pp. 45–55, 2017.

[46] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ser. ACL '09, Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 235–243, ISBN: 978-1-932432-45-9.

[47] G. de Melo and G. Weikum, "Towards universal multilingual knowledge bases," in *Proc. GWC 2010*, 2010.

[48] E. D. Gutiérrez, E. Shutova, P. Lichtenstein, G. de Melo, and L. Gilardi, "Detecting cross-cultural differences using a multilingual topic model," *TACL*, vol. 2016:4, 2016.

[49] G. de Melo and G. Weikum, "Extracting sense-disambiguated example sentences from parallel corpora," in *Proc. Workshop on Definition Extraction at RANLP 2009*, 2009.

[50] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD 2004: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, WA, USA: ACM, 2004, pp. 168–177, ISBN: 1-58113-888-1.

[51] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Comp. Int.*, vol. 29, no. 3, pp. 436–5, 2013.

[52] R. Mihalcea, C. Banea, and J. Wiebe, "Learning multilingual subjective language via cross-lingual projections," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 976–983.

[53] J. Boyd-Graber and P. Resnik, "Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10, Cambridge, Massachusetts: Association for Computational Linguistics, 2010, pp. 45–55.

[54] A. Balamurali, A. Joshi, and P. Bhattacharyya, "Cross-lingual sentiment analysis for indian languages using linked wordnets," *Proc. COLING 2012*, pp. 73–82, 2012.

[55] Y. Chen and S. Skiena, "Building sentiment lexicons for all major languages," in *Proc. ACL 2014*, 2014, pp. 383–389.

[56] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment in short strength detection informal text," *J. Am. Soc. Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.

[57] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1555–1565.

[58] O. İrsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 720–728.

[59] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *CoRR*, vol. abs/1503.00075, 2015.

[60] M. Looks, M. Herreshoff, D. Hutchins, and P. Norvig, "Deep learning with dynamic computation graphs," *CoRR*, vol. abs/1702.02181, 2017.

[61] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489.

[62] H. Chen, M. Sun, C. Tu, Y. Lin, and Z. Liu, "Neural sentiment classification with user and product attention," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1650–1659.

[63] M. Huang, Q. Qian, and X. Zhu, "Encoding syntactic knowledge in neural networks for sentiment classification," *ACM Trans. Inf. Syst.*, vol. 35, no. 3, 26:1–26:27, Jun. 2017.

[64] X. Dong and G. de Melo, "Cross-lingual propagation for deep sentiment analysis," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, AAAI Press, 2018, pp. 5771–5778.

[65] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[66] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv preprint arXiv:1410.3916*, 2014.

[67] A. Kumar *et al.*, "Ask me anything: Dynamic memory networks for natural language processing," *CoRR*, vol. abs/1506.07285, 2015.

[68] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[69] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[70] M. E. Peters *et al.*, "Deep contextualized word representations," *ArXiv e-prints*, Feb. 2018. arXiv: 1802.05365 [cs.CL].

[71] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[72] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2440–2448.

[73] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[74] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," *Proceedings of SemEval*, pp. 19–30, 2016.

[75] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd international conference on World Wide Web*, ACM, 2013, pp. 897–908.

[76] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[77] J. Blitzer, M. Dredze, F. Pereira, *et al.*, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL*, vol. 7, 2007, pp. 440–447.

[78] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[79] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," *arXiv preprint arXiv:1603.06318*, 2016.

[80] J. Choi, K. M. Yoo, and S.-g. Lee, "Unsupervised learning of task-specific tree structures with tree-lstms," *arXiv preprint arXiv:1707.02786*, 2017.

[81] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of bert," *arXiv preprint arXiv:1904.09077*, 2019.

[82] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd annual meeting of the association for computational linguistics*, 1995.

[83] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proceedings of NAACL-HLT 2006*, 2006.

[84] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," *arXiv preprint arXiv:1805.06297*, 2018.

[85] K. Clark, M.-T. Luong, C. D. Manning, and Q. V. Le, "Semi-supervised sequence modeling with cross-view training," *arXiv preprint arXiv:1809.08370*, 2018.

[86] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, ACM, 1998, pp. 92–100.

[87] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge & Data Engineering*, no. 11, pp. 1529–1541, 2005.

[88] S. Ruder and B. Plank, "Strong baselines for neural semi-supervised learning under domain shift," *arXiv preprint arXiv:1804.09530*, 2018.

[89] X. Wan, "Co-training for cross-lingual sentiment classification," in *Proceedings of ACL-IJCNLP 2009*, 2009.

[90] R. Xu and Y. Yang, "Cross-lingual distillation for text classification," *arXiv preprint arXiv:1705.02073*, 2017.

[91] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 557–570, 2018.

[92] C. Szegedy *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[93] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[94] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[95] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[96] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[97] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," *arXiv preprint arXiv:1906.02443*, 2019.

[98] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for language understanding," *arXiv preprint arXiv:1909.11764*, 2019.

[99] Z. Fu *et al.*, "Absent: Cross-lingual sentence representation mapping with bidirectional gans," *arXiv preprint arXiv:2001.11121*, 2020.

[100] S. Abney, *Semisupervised learning for computational linguistics*. Chapman and Hall/CRC, 2007.

[101] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[102] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *arXiv preprint arXiv:1206.4683*, 2012.

[103] P. Keung, Y. Lu, and V. Bhardwaj, "Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner," *arXiv preprint arXiv:1909.00153*, 2019.

[104] X. Dong and G. de Melo, "A robust self-learning framework for cross-lingual text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6307–6311.

[105] Z. Liu *et al.*, "Zero-shot cross-lingual dialogue systems with transferable latent variables," *arXiv preprint arXiv:1911.04081*, 2019.

[106] K. Yu, H. Li, and B. Oguz, "Multilingual seq2seq training with similarity loss for cross-lingual document classification," in *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, pp. 175–179.

[107] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of ACL-IJCNLP 2015*, vol. 1, 2015.

[108] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[109] X. Dong and G. de Melo, "A robust self-learning framework for cross-lingual text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov. 2019, pp. 6306–6310.

[110] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[111] A. Conneau *et al.*, "Xnli: Evaluating cross-lingual sentence representations," *arXiv preprint arXiv:1809.05053*, 2018.

[112] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[113] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[114] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," *arXiv preprint arXiv:1711.02173*, 2017.

[115] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial example generation with syntactically controlled paraphrase networks," *arXiv preprint arXiv:1804.06059*, 2018.

[116] Y. Cheng, L. Jiang, and W. Macherey, "Robust neural machine translation with doubly adversarial inputs," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4324–4333.

[117] X. Dong *et al.*, "Leveraging adversarial training in self-learning for cross-lingual text classification," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20, 2020, pp. 1541–1544.

[118] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[119] T. Luong, H. Pham, and C. D. Manning, "Bilingual word representations with monolingual quality in mind," in *Proceedings of the 1st Workshop on Vector Space*

Modeling for Natural Language Processing, Denver, Colorado, Jun. 2015, pp. 151–159.

[120] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," *CoRR*, vol. abs/1309.4168, 2013. arXiv: 1309.4168.

[121] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[122] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[123] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[124] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 644–648.

[125] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *ICLR*, 2017.

[126] Ž. Agic and I. Vulic, "Jw300: A wide-coverage parallel corpus for low-resource languages," 2020.

[127] S. Bird, "Decolonising speech and language technology," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3504–3519.

[128] M. Jaderberg *et al.*, "Reinforcement learning with unsupervised auxiliary tasks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[129] T. Trinh, A. Dai, T. Luong, and Q. Le, "Learning longer-term dependencies in rnns with auxiliary losses," in *International Conference on Machine Learning*, PMLR, 2018, pp. 4965–4974.

[130] S. Liu, A. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[131] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, Jun. 2019, pp. 4171–4186.

[132] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020.

[133] R. Han, X. Ren, and N. Peng, "Deer: A data efficient language model for event temporal reasoning," *arXiv preprint arXiv:2012.15283*, 2020.

[134] R. Han, Q. Ning, and N. Peng, "Joint event and temporal relation extraction with shared representations and structured prediction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 434–444.

[135] H. Wang, M. Chen, H. Zhang, and D. Roth, "Joint constrained learning for event-event relation extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 696–706.

[136] Y. Zhou *et al.*, "Clinical temporal relation extraction with probabilistic soft logic regularization and global inference," *arXiv e-prints*, arXiv–2012, 2020.

[137] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[138] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4487–4496.

[139] D. Xu *et al.*, "Multi-task recurrent modular networks," in *AAAI*, 2021.

[140] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[141] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, ACM, 2013, pp. 165–172.

[142] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proceedings of the ACM Conference on Recommender Systems (RecSys)*, ACM, 2014, pp. 105–112.

[143] Y. Bao, H. Fang, and J. Zhang, "Topicmf: Simultaneously exploiting ratings and reviews for recommendation," in *Twenty-Eighth AAAI conference on artificial intelligence (AAAI)*, 2014.

[144] C. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *arXiv preprint arXiv:1602.03609*, 2016.

[145] S. Wang, M. Yu, S. Chang, and J. Jiang, "A co-matching model for multi-choice reading comprehension," *arXiv preprint arXiv:1806.04068*, 2018.

[146] M. E. Peters *et al.*, "Deep contextualized word representations," in *NAACL-HLT*, 2018, pp. 2227–2237.

[147] X. Dong and G. De Melo, "A helping hand: Transfer learning for deep sentiment analysis," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 2524–2534.

[148] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[149] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, and Y. Zhang, "Reinforcement knowledge graph reasoning for explainable recommendation," in *Proceedings of SIGIR 2019*, (Jul. 21, 2019), Paris, France: ACM, 2019, pp. 285–294, ISBN: 978-1-4503-6172-9.

[150] C. Wu, F. Wu, J. Liu, and Y. Huang, "Hierarchical user and item representation with three-tier attention for recommendation," in *NAACL-HLT*, 2019, pp. 1818–1826.

[151] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, ACM, 2018, pp. 2309–2318.

[152] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[153] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2132–2141.

[154] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," in *ICLR*, 2017.

[155] X. Zhang, S. Li, L. Sha, and H. Wang, "Attentive interactive neural networks for answer selection in community question answering," in *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[156] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems (NIPS)*, 2008, pp. 1257–1264.

[157] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems (NIPS)*, 2001, pp. 556–562.

[158] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[159] X. Dong *et al.*, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 7667–7674.

[160] X. Dong, Y. Zhu, Z. Fu, D. Xu, and G. de Melo, "Data augmentation with adversarial training for cross-lingual nli," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 5158–5167.

[161] X. Dong, T. K. Saha, K. Zhang, J. Tetreault, A. Jaimes, and G. de Melo, "Temporal event reasoning using multi-source auxiliary learning objectives," in *European Conference on Information Retrieval*, Springer, 2022, pp. 102–110.