Effective Training with Data Engineering for Language Understanding and Generation

JIAO, Wenxiang

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
September 2021

Thesis Assessment Committee

Professor YIP Yuk Lap (Chair)

Professor KING Kuo Chin Irwin (Thesis Supervisor)

Professor LYU Rung Tsong Michael (Thesis Co-supervisor)

Professor LIU Xunying (Committee Member)

Professor LIN Shou-De (External Examiner)

Abstract of thesis titled:

Effective Training with Data Engineering for Language Understanding and Generation

Submitted by JIAO, Wenxiang

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in September 2021

Existing natural language processing (NLP) systems are built on three basic components, namely, data, model, and algorithm. Among the three, data is the foundation of NLP systems, since it decides the architecture of models, the scale of models, and the corresponding optimization algorithms. Thus, it has been critical to keep improving the training effectiveness on data for better performance. In this thesis, we present our exploitation of data in two dimensions, i.e., the intra-sample structure and the inter-sample quality. We define intra-sample structure exploitation as fully utilizing the structure information shared by all text samples while inter-sample quality exploitation as emphasizing a certain type of text samples. Regarding intra-sample structure exploitation, we work on the emotion recognition in conversations (ERC) task due to the rich structure information in conversations and focus on context enhancement and self-supervised learning. For inter-sample quality, we work on neural machine translation (NMT) which is a classic language generation task with large-scale benchmark datasets and complete evaluation criteria, and focus on uncertainty-based data rejuvenation and self-training sampling.

First, we investigate the intra-sample structure for context enhancement in conversations. It is important to capture the context information accurately to perform the ERC task as an utterance

may express different emotions in different contexts. Therefore, we propose a hierarchical gated recurrent unit (HiGRU) framework with a lower-level GRU to capture word-level contexts and an upper-level GRU to capture utterance-level contexts. We further promote the vanilla HiGRU to two variants, HiGRU with individual features fusion (HiGRU-f) and HiGRU with self-attention and features fusion (HiGRU-sf), so that the word- and utterance-level individual inputs and the long-range context information can be sufficiently utilized. Experiments on three widely used ERC datasets demonstrate that our HiGRU models are effective in improving the performance of ERC.

Second, we study the intra-sample structure to perform self-supervised learning on unlabeled conversation data. Data scarcity is an issue for the ERC task, which prevents the commonly used context-dependent models from playing their maximum effect. However, annotating more data is costly and difficult as annotators are required to recognize the subtle difference between emotions and consider the contexts. Therefore, we propose a conversation completion (ConvCom) task to pre-train a context-dependent encoder (Pre-CODE) on the massive unlabeled conversation data. Essentially, such a pre-training task utilizes the sequential relationship between utterances in each conversation, enabling the learning of representations at both utterance- and conversation-level. Through fine-tuning on labeled data, our Pre-CODE achieves constantly significant improvements across the ERC datasets.

Third, we exploit the inter-sample quality for uncertainty-based data rejuvenation on the NMT task with large-scale datasets. Large-scale datasets are important for training large and well-performing models, but also contain complex patterns and potential noises that pose challenges for training NMT models effectively. Thus, we identify the inactive data in a dataset which contributes less to the model performance by model uncertainty, and show that the existence of inactive data depends on the data distribution. We further introduce data rejuvenation (DataReju) to improve the training of NMT models on large-scale datasets by re-labeling the inactive data with

forward-translation. Experiments on large-scale datasets show that our DataReju approach consistently and significantly improves the performance of strong NMT models.

At last, we explore the inter-sample quality for uncertainty-based self-training sampling to leverage monolingual data more efficiently for the NMT task. Self-training augments the training of NMT models with synthetic parallel data, in which the target sentences are obtained by translating the monolingual sentences at the source side. The monolingual sentences usually come from a randomly sampled subset of large-scale monolingual data. However, we empirically show random sampling is sub-optimal and propose to select the most informative monolingual sentences for self-training. Thus, we calculate the translation uncertainty of monolingual sentences based on a bilingual dictionary, and propose an uncertainty-based sampling (UncSamp) strategy that prefers to sample monolingual sentences with relatively higher uncertainty for self-training. Experiments on large-scale NMT datasets demonstrate that our UncSamp strategy improves the translation quality significantly, especially for uncertain sentences.

論文題目：語言理解與生成中基於數據工程的模型有效訓練

作者　　：焦文祥

學校　　：香港中文大學

學系　　：計算機科學與工程學系

修讀學位：哲學博士

摘要　　：

當下，先進的自然語言處理（NLP）系統都是基於三個基礎要素，即，數據、模型和算法。在這三個要素中，數據是NLP系統的根本，因為它決定了所使用的模型框架、模型規模和對應的優化算法。因而，如何持續提高在數據上的訓練有效性以獲得更好的性能是一個十分重要的問題。在本論文中，我們從兩個維度展示了我們對於數據的開發利用，即，樣本內結構和樣本間質量。我們定義樣本內結構開發為充分地利用所有樣本共享的結構信息，而樣本間質量開發為强調對某一類樣本的學習。對於樣本內結構開發，考慮到對話具有豐富的結構信息，我們在一個基於對話的情緒識別（ERC）任務上進行研究並關注於兩個問題，包括上下文增强和自監督學習。對於樣本間質量開發，我們在神經機器翻譯（NMT）任務上開展研究，該任務是一個經典的語言生成任務，並且具有大規模訓練數據和完備的評價體系。我們關注兩個問題，包括基於不確定性的數據重生和基於不確定性的自訓練採樣。

首先，我們通過開發文本對話樣本內結構來獲得上下文增强，從而提升模型在ERC任務上的效果。對於ERC任務來説，如何準確地捕捉上下文信息十分重要，因為一句話在不同的上下文裡可能表達不同的情緒。因此，我們提出了一種層級結構的門控循環單元（HiGRU），其包括一個用於捕捉單詞級別上下文的GRU和一個用於捕捉句子級別上下文的GRU。進一步，我們提出了HiGRU兩

個改進版本，即，具有個體特征融合的HiGRU（HiGRU-f）和同時具有個體特征融合與自注意力機制的HiGRU（HiGRU-sf），充分地利用了單詞/句子級別個體特征和長距離上下文信息。通過在三個常用的ERC數據集上開展實驗，我們驗證了所提出的HiGRU模型能夠有效地提升預測性能。

其次，我們通過開發文本對話樣本內結構來進行自監督學習，從而利用無標籤的文本對話數據。數據稀缺是ERC任務的一個重要問題，它阻礙了常用的上下文依賴模型充分發揮效果。然而，人工標註更多數據昂貴且困難，因為標註人員需要辨別不同情緒類別間微弱的差異，以及需要考慮所處的上下文。因此，我們提出了一個對話補全（ConvCom）任務，用於在大量的無標籤對話數據上預訓練一個依賴上下文的編碼器（Pre-CODE）。本質上，這樣的預訓練任務利用了每個對話裡句子之間的序列關係，因而可以同時在句子和對話量級進行表示學習。通過在有標籤數據上進行精調，我們提出的Pre-CODE方法能夠顯著且一致地提升ERC任務的效果。

接著，我們通過開發NMT大規模數據的樣本間質量來進行基於不確定性的數據重生。大規模數據是訓練一個性能良好的大規模NMT模型的要素，但其中的複雜模式及潛在噪聲也給模型的有效訓練帶來的挑戰。因而，我們提出通過模型的不確定性來識別訓練數據的不活躍數據，即那些對模型性能貢獻很小的數據，並且驗證了不活躍樣本的存在主要和數據分佈本身有關。進一步，我們提出了數據重生（DataReju），通過正向翻譯的方式對不活躍數據進行重新打標籤，從而提升NMT模型在大規模數據上的訓練效果。通過在兩個大規模NMT數據集上進行實驗，我們驗證了DataReju能夠在强悍的基線模型基礎上得到一致且顯著的提升。

最後，我們通過開發大規模單語數據的樣本間質量來進行基於不確定性的自訓練採樣，從而更加高效地利用單語數據提升NMT任務的性能。自訓練能夠給NMT模型補充偽平行數據，即目標端句子是通過源端單語句子翻譯得到的。這些單語句子通常是來自於大規模單語數據中隨機採樣的一個子集。然而，我們通過實驗證明了隨機採樣並不是最優的，並提出選擇最具有信息量的單語數據來開展自訓練。具體地，我們基於一個從真實平行語料中抽取的雙語詞典，計算了每個單語句子的翻譯不確定性。接著，我們提出了基於

不確定性的採樣（UncSamp）策略，在採樣時更傾向於選擇不確定性更高的單語數據。在大規模NMT數據集上的實驗顯示，我們的UncSamp方法能夠顯著提高翻譯質量，尤其是對於不確定的句子的翻譯，並且我們的方法能提升目標端低頻詞的預測準確率。

Acknowledgement

First and foremost, I would like to thank my supervisors, Prof. Irwin King and Prof. Michael R. Lyu, for their excellent supervision during my Ph.D. study at CUHK. On one hand, their open mind allows me to explore interesting research topics without hesitation. On the other hand, their rigorous attitude towards research deeply influences me and prompts me to improve myself continuously. During the long Ph.D. study period, I have learned so much from them, not only their knowledge in research but also their wisdom of life.

I am very grateful to my thesis assessment committee members, Prof. Kevin Yip and Prof. Andrew Xunying Liu, for their constructive comments and insightful suggestions to this thesis and all the term presentations during my Ph.D. study. Great thanks to Prof. Shou-De Lin from National Taiwan University, who kindly serves as the external examiner for this thesis.

I would like to thank my mentors, Dr. Xing Wang, and Dr. Zhaopeng Tu when I interned in Tencent AI Lab for their valuable contributions to the research in this thesis. Also, I would like to thank my colleagues during the internship, Longyue Wang, Yong Wang, Yilin Yang, Bo He, Boyuan Wang, Wenxuan Wang, Shuo Wang, Cunxiao Du, Xuebo Liu, Liang Ding, Mingzhou Xu, Hongye Liu, Yongchang Hao, Zhiwei He, and Jiaqing Zhang for their great help in my research and life.

I am very thankful to my fantastic group fellows, Haiqin Yang, Hongyi Zhang, Shenglin Zhao, Xiaotian Yu, Yuxin Su, Cuiyun Gao, Jichuan Zeng, Jiani Zhang, Ken Chan, Jian Li, Han Shao, Wang Chen, Yue Wang, Pengpeng Liu, Shilin He, Haoli Bai, Yifan Gao, Weibin

Wu, Jingjing Li, Zhuangbin Chen, Tianyi Yang, Ziqiao Meng, and Wenchao Gu, who are the family of mine in CUHK.

I also want to thank some old friends during my study at Nanjing University who have always been supporting me and become an important part of my life. They are my roommates, Xiaolei Li, Gaozheng Jin, Jian Yuan, Yonghao Zhao, Ziqiao Luan, Guanqun Zhou, and my good friends, Dongdong Chen, Jia Gu, Jieyu Ding, and Zhifei Ding.

Last but most importantly, I would like to thank my family. Thanks to my girlfriend, Miss Yuye Wang, who taught me how to enjoy life. Her unreserved love, meticulous care, and constant companionship are the great motivation for me to complete my Ph.D. study. Thanks to my parents, my sister, my brother-in-law, my grandparents, and all my other family members. Their deep love and unconditional trust are the driving force for me to thrive. Special thanks to my little nephew and niece. Being with these two cute kids always makes me feel younger and more energetic.

Contents

Abstract i

Acknowledgement vii

1 Introduction 1
- 1.1 Overview . . . 1
- 1.2 Thesis Contributions . . . 9
- 1.3 Publications During Ph.D. Study . . . 11
- 1.4 Thesis Organization . . . 12

2 Background Review 15
- 2.1 Context Modeling . . . 15
  - 2.1.1 Sequential Modeling . . . 16
  - 2.1.2 Hierarchical Modeling . . . 21
- 2.2 Self-Supervised Learning . . . 27
  - 2.2.1 Word-Level Pre-Training . . . 27
  - 2.2.2 Sentence-Level Pre-Training . . . 29
  - 2.2.3 Hierarchical Pre-Training . . . 33
- 2.3 Data Re-Weighting . . . 34
  - 2.3.1 Dynamic Weighting . . . 34
  - 2.3.2 Data Selection . . . 36
  - 2.3.3 Curriculum Learning . . . 37
- 2.4 Data Augmentation . . . 38
  - 2.4.1 Back-Translation . . . 38
  - 2.4.2 Self-Training . . . 39
  - 2.4.3 Hybrid Approach . . . 41
- 2.5 Inference and Evaluation . . . 42

2.5.1 Datasets . . . 42
2.5.2 Inference . . . 47
2.5.3 Evaluation Metrics . . . 47

3 Intra-Sample Structure Mining for Context Enhancement 51
3.1 Problems and Motivation . . . 51
3.2 Methodology . . . 54
3.2.1 Vanilla HiGRU . . . 54
3.2.2 Individual Features Fusion . . . 56
3.2.3 Long-Range Context . . . 57
3.2.4 Training Objective . . . 58
3.3 Experiment . . . 59
3.3.1 Datasets . . . 59
3.3.2 Evaluation Metrics . . . 61
3.3.3 Compared Methods . . . 61
3.3.4 Training Procedure . . . 62
3.3.5 Main Results . . . 63
3.4 Analysis . . . 66
3.4.1 Model Size . . . 66
3.4.2 Successful Cases . . . 67
3.4.3 Failed Cases . . . 69
3.5 Summary . . . 70

4 Intra-Sample Structure Mining for Self-Supervised Learning 71
4.1 Problems and Motivation . . . 72
4.2 Methodology . . . 73
4.2.1 Pre-training Task . . . 73
4.2.2 Pre-training Model . . . 74
4.3 Experiment: Pre-training . . . 76
4.3.1 Unlabeled Conversation Data . . . 76
4.3.2 Noise Utterances . . . 77
4.3.3 Evaluation . . . 77
4.3.4 Training Details . . . 78

4.3.5 Results . . . 78
4.4 Experiment: Fine-tuning . . . 79
4.4.1 ERC Architecture . . . 79
4.4.2 Compared Methods . . . 79
4.4.3 ERC Datasets . . . 80
4.4.4 Evaluation . . . 82
4.4.5 Training Details . . . 82
4.4.6 Results . . . 83
4.5 Analysis . . . 84
4.5.1 Model Capacity . . . 84
4.5.2 Layer Effect . . . 84
4.5.3 Qualitative Study . . . 85
4.6 Summary . . . 86

5 Inter-Sample Quality Mining for Uncertainty-Based Data Rejuvenation 89
5.1 Problems and Motivation . . . 90
5.2 Methodology . . . 92
5.2.1 Identification Model . . . 93
5.2.2 Rejuvenation Model . . . 93
5.3 Experiment . . . 94
5.3.1 Experimental Setup . . . 94
5.3.2 Identification of Inactive Samples . . . 95
5.3.3 Rejuvenation of Inactive Samples . . . 99
5.3.4 Main Results . . . 100
5.4 Analysis . . . 104
5.4.1 Linguistics Properties . . . 105
5.4.2 Learning Stability . . . 106
5.4.3 Generalization Capability . . . 107
5.4.4 Speeding Up . . . 108
5.4.5 Inactive Sample Cases . . . 109
5.5 Summary . . . 110

6 Inter-Sample Quality Mining for Uncertainty-Based Self-Training Sampling 113

6.1 Problems and Motivation . . . 114
6.2 Preliminary . . . 116
6.2.1 Identification of Uncertain Data . . . 116
6.2.2 Experimental Setup . . . 117
6.2.3 Effect of Uncertain Data . . . 119
6.2.4 Linguistic Properties of Uncertain Data . . . 121
6.3 Methodology . . . 123
6.3.1 Uncertainty-based Sampling Strategy . . . 123
6.3.2 Overall Framework . . . 124
6.4 Experiment . . . 125
6.4.1 Constrained Scenario . . . 125
6.4.2 Unconstrained Scenario . . . 128
6.5 Analysis . . . 129
6.5.1 Uncertain Sentences . . . 129
6.5.2 Low-Frequency Words . . . 130
6.6 Summary . . . 131

7 Conclusion and Future Work **133**
7.1 Conclusion . . . 133
7.2 Future Work . . . 135
7.2.1 Low-Frequency Issue in Data Augmentation . . . 135
7.2.2 Low-Frequency Issue in Multilingual Machine Translation . . . 136
7.2.3 Self-Supervised Multilingual Pre-training . . . 137

Bibliography **139**

List of Figures

1.1 A general diagram for text classification. 2
1.2 An example of emotion recognition in conversations from *Friends*. 5
1.3 An example of machine translation by Google Translate. 7
1.4 Overview of the research in this thesis. 8

2.1 Overview of the research on effective training with data engineering. Cells marked with publication venues (e.g., AAAI'20) represent our studies, among which the venues in red are included in this thesis. 16
2.2 The architecture of one layer CNN with two channels for a text sample [1]. 17
2.3 The unfold recurrent neural network [2]. 18
2.4 The architecture of SAN [3]. 20
2.5 The architecture of HAN [4]. 22
2.6 The architecture of cLSTM [5]. 23
2.7 The architecture of CMN [6]. 25
2.8 The architecture of DialogueRNN [7]. 26
2.9 The architectures of Skip-gram and Continuous Bag of Words models. 28
2.10 The architecture of ELMo [8]. 30
2.11 The architecture of GPT [8]. 31
2.12 The architecture of BERT [8]. 32
2.13 The architecture of TL-ERC [9]. 33
2.14 Overview of the competence-based curriculum learning [10]. 37
2.15 Workflows of back-translation and self-training. 40

2.16 A confusion matrix of binary classification. 48

3.1 The architecture of our proposed HiGRU-sf. "Attention" denotes self-attention. By removing the "Attention" layer, we attain HiGRU-f, and by further removing the "Fusion" layer, we can recover the vanilla HiGRU. . 54

3.2 Self-attention over the forward hidden states of GRU. . 58

4.1 A data example in the ConvCom task. 74

4.2 The architecture of the context-dependent encoder with the pre-training objective. 75

4.3 The architecture for the ERC task. Both the utterance encoder and conversation encoder are transferred from the PRE-CODE. 80

4.4 Detailed F1-score of each emotion class on IEMOCAP and EmoryNLP. 83

5.1 The framework of data rejuvenation. The inactive samples from the original training data are identified by the *identification model*, then rejuvenated by the *rejuvenation model*. The rejuvenated samples along with the active samples are used together to train the NMT model. 92

5.2 Probability diagram on (a) En⇒De and (b) En⇒Fr datasets. Training samples in smaller bins (e.g., 1, 2) are regarded as inactive samples due to their lower probabilities. 95

5.3 Translation performance of the NMT model trained on the training data with the most inactive samples removed. For comparison, results of the most active samples and randomly sampled samples are also presented. 96

5.4 Ratio of samples that are shared by different model variants: random seed (a), model capacity (b), model architecture on En⇒De (c) and En⇒Fr (d) datasets. A high overlapping ratio for most inactive samples (i.e., 1^{st} data bin) demonstrates that the identified inactive samples are not model-specific. 98

5.5 Effect of the ratio of samples labeled as inactive samples. We used forward-translation as the rejuvenation strategy and trained the final NMT model on the combination of rejuvenated samples and active samples from scratch. 99

5.6 Linguistic properties of different training samples: frequency rank (↑ more difficult), coverage (↓ more difficult), and uncertainty (↑ more difficult). 105

5.7 Learning curves on the En⇒De dataset. 107

5.8 Probability and ratio of source-translated samples over the data bins of En⇒De test set. 109

6.1 Performance of self-training with increased size of monolingual data. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020. 120

6.2 Relationship between uncertainty of monolingual data and the corresponding NMT performance. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020. 120

6.3 Comparison of monolingual sentences with varied uncertainty in terms of three properties, including sentence length, word rarity, and coverage. 121

6.4 Distribution of modified monolingual uncertainty and sampling probability. The sample with high uncertainty has more chance to be selected while that with excessively high uncertainty would be penalized. 123

6.5 Framework of the proposed uncertainty-based sampling strategy for self-training. Procedures framed in the red dashed box corresponds to our approach integrated into the standard self-training framework. "Bitext", "Mono", "Synthetic" denotes authentic parallel data, monolingual data and synthetic parallel data, respectively. 125

List of Tables

2.1 Datasets for emotion recognition in conversations. . . . 43
2.2 Datasets for machine translation. "⇒" denotes the translation task in the forward direction and "⇐" in the backward direction. 45

3.1 The word "okay" exhibits different emotions in the American television sitcom, Friends. 52
3.2 Statistics of the textual conversation datasets. 59
3.3 Experimental results on IEMOCAP. "(Feat)" represents the features used in the models, where T, V, and A denote the textual, visual, and audio features, respectively. The underlined results of bcLSTM and CMN are derived by us accordingly, while "-" represents that the results are unavailable from the original paper. For each emotion class, the corresponding column of values are the accuracy scores. 63
3.4 Experimental results on Friends and EmotionPush. In the Train column, F(E) denotes the model is trained on only one training set, Friends or EmotionPush. F+E means the model is trained on the mixed training set while validated and tested individually. 64
3.5 Experimental results of UWA on Friends by our proposed models with different scales of utterance encoder. 67
3.6 "Okay" expresses distinct emotions in three different scenes. 68
3.7 Wrong predictions made by both bcGRU and our HiGRU-sf in two scenes. 69

4.1 Statistics of the datasets for ERC. 72
4.2 Statistics of the created datasets for the ConvCom task. 77
4.3 Test results of CODE on the ConvCom task in three capacities. 79
4.4 Test results on IEMOCAP, EmoryNLP, and MOSEI*. The implemented bcLSTM performs much better than the original one, possibly because that the original bcLSTM is not trained end-to-end. 81
4.5 Test results on Friends and EmotionPush. 82
4.6 Ablation study on model capacity. 85
4.7 Ablation study on pre-trained layers. 85
4.8 Qualitative comparison between CODE and PRE-CODE by two examples. 86

5.1 Effect of different rejuvenation strategies. 100
5.2 Comparing data rejuvenation on identified inactive samples and forward translation on randomly sampling samples. 101
5.3 Evaluation of translation performance across model architectures and language pairs. "↑ / ⇑": indicate statistically significant improvement over the corresponding baseline $p < 0.05/0.01$ respectively. 102
5.4 Comparison with other data manipulation approaches. Results are reported on the En⇒De test set. 102
5.5 New testing rule on WMT19 En⇒De datasets, evaluated on newstest2019 and newstest2020. 104
5.6 Results of generalization capability on the En⇒De dataset. Larger Margin and GSNR values denote better generalization capability. 108
5.7 Results of speeding up ("Rej.–Big") on the WMT14 En⇒De dataset. "Time" denotes the time of the whole process using 4 NVIDIA Tesla V100 GPUs. 109

5.8 Inactive samples from the training sets of En⇒De and En⇒Fr. `X`, `Y` and `Y'` represent the source sentence, target sentence, and the rejuvenated target sentence, respectively. `Y` and `Y'` are also translated into English (`=>En:`) by Google Translate for reference. For either sample, the underlined phrases correspond to the same content. 111

6.1 Translation performance with respect to different values of β and R. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020. 126

6.2 Comparison of our UncSamp and RandSamp with manual translations (Ora: manual translations; ST: pseudo-sentences) on WMT En⇒De newstest2019 and newstest2020. 127

6.3 Comparison of the proposed uncertainty-based sampling strategy with related methods on WMT En⇒De newstest2019 and newstest2020. 127

6.4 Translation performance on WMT En⇒De and WMT En⇒Zh test sets. The results are reported with de-tokenized case-sensitive SacreBLEU. We adopt the Transformer-big with large batch training [11] to achieve the strong performance. "↑ / ⇑": indicate statistically significant improvement over RandSamp $p < 0.05/0.01$ respectively. 128

6.5 Translation performance on uncertain sentences. The relative improvements over Bitext for UncSamp are also presented. 130

6.6 Prediction accuracy of low-frequency words in the translation outputs. The relative improvements over Bitext for UncSamp are also presented. 131

Chapter 1

Introduction

This thesis presents our research on effective training with data engineering for language understanding and generation. We first provide a brief overview of the research problems explored in Section 1.1 and highlight the main contributions of this thesis in Section 1.2. Then we list the publications that are related to this thesis during my Ph.D. study in Section 1.3 and outline the thesis structure in Section 1.4.

1.1 Overview

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence (AI), aiming at extracting insights from natural language texts or even generating new texts with computers[1]. Like other AI areas, existing NLP systems are built on three basic components, namely, data, model, and algorithm, just as Andrew Ng said[2]. Data is the foundation of NLP systems, since it decides the architecture of models, the scale of models and the corresponding optimization algorithms. For example, language understanding tasks are usually based on text-label pairs[3] while language generation tasks are based on text-text pairs[4]. The different types of input data determine that the former tasks can be addressed

[1] https://en.wikipedia.org/wiki/Natural_language_processing
[2] https://www.deeplearning.ai/programs/
[3] http://nlpprogress.com/english/sentiment_analysis.html
[4] http://nlpprogress.com/english/machine_translation.html

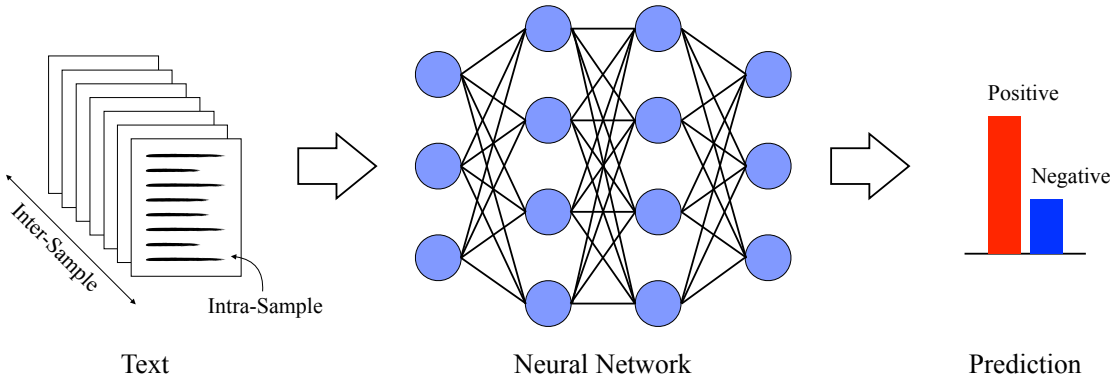


Figure 1.1: A general diagram for text classification.

by an encoder-classifier architecture [1, 12] while the latter by an encoder-decoder architecture [13, 14, 15]. Besides, tasks with small-scale data should be tackled by small models, otherwise the models will be over-fitting to the training data and cannot generalize well to the test data. For example, the Transformer model [15] used for IWSLT14 German-English translation task is configured with only 1/2 parameters[5] of the Transformer-base model for WMT14 English-German translation task due to the much smaller dataset (160K vs. 4.5M sentence pairs). For large-scale data, large models are usually preferred to memorize the knowledge from the data as much as possible, for example, the Transformer-big model for WMT14 English-French translation tasks [16] and deeper Transformer models [17]. In addition, it is also critical to design how to feed the data [10, 18] or whether certain kinds of data should be emphasized [19, 20, 21]. Obviously, how to train models more effectively on the data to achieve better performance on NLP tasks is a significant problem for researchers.

Figure 1.1 shows a general diagram of a text classification task implemented by a neural network. To categorize, the text data can be exploited in two different dimensions, i.e., fully utilizing the shared structure information in each text sample or differentiating the text samples by their quality. We define the first as **intra-sample structure exploitation** and the second as **inter-sample quality**

[5] https://github.com/pytorch/fairseq/blob/master/fairseq/models/transformer.py

exploitation. As defined, intra-sample structure exploitation focuses on the structure information shared by all text samples. For examples, a document or conversation is composed of multiple sentences, each of which further contains a sequence of words. Such a hierarchical structure conforms to any document or conversation, providing rich information for representation learning. On the contrary, inter-sample quality exploitation tries to find the quality difference among the text samples and give different importance to the text samples accordingly during training. The quality of samples could be measured by some metrics such as the model confidence or linguistic properties (e.g., sentence length, and word rarity). Both directions are capable of improving the effectiveness of training and the final performance.

In terms of intra-sample structure, the data structure information provides important clues for representation learning in NLP. The development of representation learning in recent years exactly reflects the process of exploiting the data structure information in texts. For example, as one of the NLP milestones, word embeddings from Word2Vec [22] or GloVe [23] are obtained by learning the local relationship between each word and its neighbors in the texts, which gathers words with similar semantics into the same clusters in vector space. However, these word embeddings are weak at modeling polysemy (i.e., the coexistence of many possible meanings for a word or phrase) as they are learned without considering the context in the entire sentences. To address such a problem, ELMo [24] has been proposed to learn deep contextualized word representations, which are functions of internal states of a deep bidirectional language model pre-trained on a large text corpus. The advantage of ELMo over Word2Vec or GloVe is that it utilizes the sequential relationship between words in a sentence, modeling better the data structure of each sentence. Following EMLo are the famous pre-trained models based on Transformer, including the BERT familiy [8, 25] and the GPT family [26, 27, 28]. The main difference between the two families is that BERT captures the context of a word by the words both before and after it with a masked language model, whereas GPT insists on

a traditional causal language model and keeps increasing the scales of both model and text corpus. With a similar model capacity, BERT is supposed to perform better than GPT on downstream tasks, of which we attribute the reason to the exploitation of bidirectional contexts. In other words, BERT exploits the data structure information of sentences more thoroughly than GPT [8].

As introduced above, the studies on intra-sample structure has mainly been conducted for sentences, which may contain less structure information than more complex text formats like documents and conversations. In this thesis, we present our exploitation of intra-sample structure information in conversations for a prevalent language understanding task, i.e., emotion recognition in conversations (ERC) [5, 6]. We choose the ERC task because of the rich structure information in conversations and the relatively easy-to-follow implementations of the models. The ERC task is defined as below:

Definition 1 (ERC) *Suppose we are given a set of conversations,* $\mathcal{D} = \{D_i\}_{i=1}^{L}$*, where* L *is the number of conversations. In each conversation,* $D_i = \{(\mathbf{x}_j, s_j, c_j)\}_{j=1}^{N_i}$ *is a sequence of* N_i *utterances, where the utterance* $\mathbf{x}_j$ *is spoken by the speaker* $s_j \in \mathcal{S}$ *with a certain emotion* $c_j \in \mathcal{C}$*. All speakers compose the set* $\mathcal{S}$ *and the set* $\mathcal{C}$ *consists of all emotions, such as anger, joy, sadness, and neutral. Our goal is to train a model* $p(c|\mathbf{x};\Theta)$ *to tag each new utterance with an emotion label from* $\mathcal{C}$ *as accurately as possible. Here,* Θ *represents the parameters of the model. Figure 1.2 shows an example from the TV sitcom Friends*[6]*.*

Accordingly, we investigate the intra-structure information for the ERC task in two aspects: *context enhancement*, and *self-supervised learning*, which are elaborated as below:

- **Context enhancement.** We hope to fully utilize the structure information in conversations to learn accurate representations such that the ERC task can be well accomplished. While the task can be performed for each individual utterance as

[6] https://www.imdb.com/title/tt0108778/

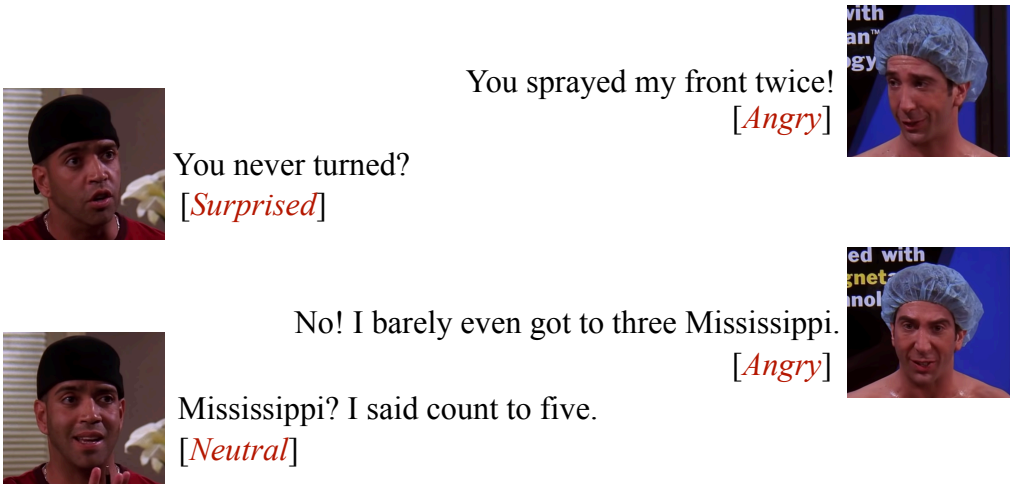


Figure 1.2: An example of emotion recognition in conversations from *Friends*.

traditional sentence-level text classification tasks, the context between utterances also plays a critical role as an utterance may indicate different emotions in different contexts.

- **Self-supervised learning.** We also attempt to leverage unlabeled conversation data with self-supervised learning by utilizing the structure information. This is particularly important for tasks like ERC that face the data scarcity issue. Usually, labels provide supervision for the training of models, which however are not available for unlabeled data. To enable the training on unlabeled data, it is important to explore the structure information in conversations for self-supervised learning signals. These signals could be the sequential relationship between utterances in each conversation, which can be used for pre-training at both sentence-level and conversation-level.

The above issues are representative since: 1) Recognizing emotions in conversations is a novel direction of sentiment analysis, which uses more practical user-generated texts. 2) Exploiting the intra-sample structure information is especially critical for small-scale datasets, which is exactly what the ERC task faces. 3) Leveraging unlabeled data is the trend of NLP (also for other areas), which inevitably needs the structure information to provide self-supervised signals.

In terms of inter-sample quality, it is especially important for large-scale datasets. Nowadays, it has become a fashion to increase the scale of datasets in order to train larger models [28]. Large datasets are usually collected through automatic methods [29] rather than human annotations. Without human correction, the source of data could be heterogeneous and the data could be noisy. In distinguishing the text sample quality, there have been a number of strategies to take advantage of data, including data re-weighting, data selection, and curriculum learning. Data re-weighting assigns higher weights to preferred types of data when calculating the loss function. There are mainly three choices of preferred data: 1) self-paced learning [19] that prefers easy samples, 2) hard sample mining [20] that exploits hard samples and 3) active learning [21] that emphasizes high-variance samples. As for data selection, it could be considered as an offline version of data re-weighting. It selects the preferred types of data and feeds it to the models, without changing the training algorithm. Commonly used metrics for data selection could be based on language models [30], low-frequency words [31], or model uncertainty [32]. Curriculum learning [10, 33, 34, 35] suggests that the training of models may follow the learning process of human, and proposes to feed the data from easy to difficult such that the models can converge faster. The metrics for measuring the difficulty of text samples could be sentence length and word rarity [10], data and model uncertainty [36], or embedding norm [18]. Generally, exploiting inter-sample quality information can accelerate the training of models and achieve a significant boost of performance.

As introduced above, previous studies on inter-sample quality only emphasize or select the preferred data but discard the rest. We are curious about whether there is a way to re-use the discarded data and thus fully utilize the whole data. Besides, we also hope to leverage large-scale unlabeled data more efficiently based on the inter-sample quality information. In this thesis, we investigate the inter-sample quality on a classic language generation task, i.e., machine translation (MT), for the large-scale benchmark datasets and its

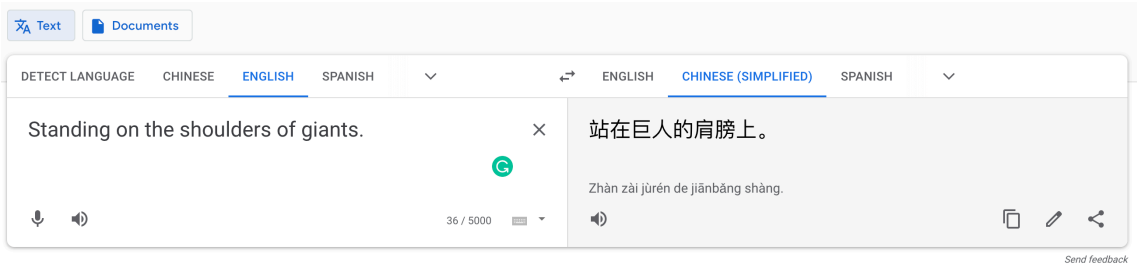


Figure 1.3: An example of machine translation by Google Translate.

complete evaluation criteria.

Definition 2 (MT) *Suppose we are given a set of sentence pairs from two languages X and Y, $\mathcal{D} = \{[\mathbf{x}_i, \mathbf{y}_i]\}_{i=1}^{N}$, where N is the number of sentence pairs, $\mathbf{x}_i$ and $\mathbf{y}_i$ are sentences from X and Y. Our goal is to train a translation model $p(\mathbf{y}|\mathbf{x}, \Theta)$ based on these sentence pairs, which can translate each new source sentence into a semantically equivalent and fluent sentence in the target language. In this translation model, X is called the source language, Y is the target language, and Θ denotes the parameters of the model. Figure 1.3 shows an translation example by Google Translate*[7].

In particular, we focus on data selection for both large-scale bilingual data and monolingual data (i.e., labeled and unlabeled data). While MT usually involves multiple languages, researchers have been trying to utilize the monolingual data to boost the training of MT models, for example, through pre-training [37] or data augmentation [38]. Accordingly, we tackle two specific issues: *Uncertainty-based Data Rejuvenation*, and *Uncertainty-based Self-training Sampling*, which are elaborated as below:

- **Uncertainty-based Data Rejuvenation.** We aim to identify samples in the training data that contribute little to the performance of models, which we define as inactive data. The criterion for identification is based on the data uncertainty calculated as the output probability of pre-trained MT models. Beside identification, we hope to understand the characteristics of inactive data and re-activate it to boost the performance of the

[7] https://translate.google.com/

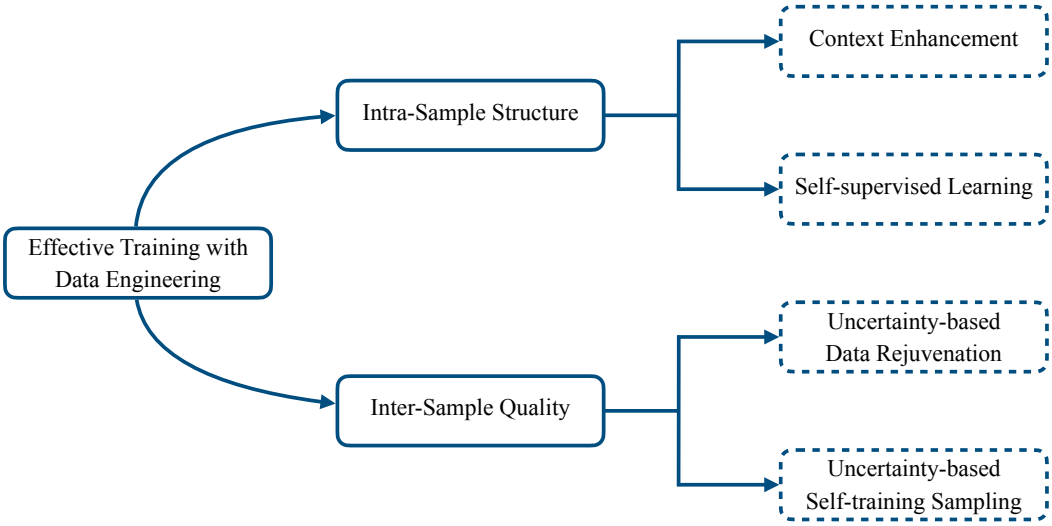


Figure 1.4: Overview of the research in this thesis.

models. The inactive data could be noisy data or sentences with complex expressions, which may reduce the training efficiency of the models.

- **Uncertainty-based Self-training Sampling.** We hope to leverage the large-scale monolingual data more efficiently when performing self-training for the MT task. Instead of translating all monolingual data indiscriminately to construct the synthetic parallel data, we attempt to sample monolingual data that are both informative and with high-quality translations. The metric for measuring the informativeness of monolingual sentences could be translation uncertainty with a bilingual dictionary. We propose an uncertainty-based sampling strategy to prefer monolingual sentences with relatively high uncertainty.

The above issues are representative since: 1) The data (both bilingual and monolingual) for current MT systems has been scaled significantly, which inevitably contains inactive samples. 2) Inactive data may contain knowledge that is not covered by the active data, which should be re-used for better generalization capability. 3) Monolingual data is massive but cannot be completely translated due to the resource limit, which requires an economically alternative approach.

Therefore, the research of this thesis comprises two parts, as illustrated in Figure 1.4. In the first part, to improve the model training effectiveness, we investigate the intra-sample structure information in conversations for emotion recognition. This part consists of two issues: context enhancement and self-supervised learning. In the second part, we aim to improve the training effectiveness by exploiting inter-sample quality information for machine translation. This part also focuses on two issues: uncertainty-based data rejuvenation and uncertainty-based self-training sampling. Generally, this thesis mainly focuses on improving the effectiveness of model training from the angle of data engineering, involving a small part of model design. For the optimization algorithms, we follow the default settings used in previous studies although we believe there is also room for improvement. Besides, while it is also possible to incorporate the well-known grammars of languages or some other prior knowledge, which may require manual feature engineering, we focus on the end-to-end training strategy to learn these features automatically in this thesis.

1.2 Thesis Contributions

In this thesis, we mainly focus on effective training of NLP models from the data engineering perspectives including intra-sample structure and inter-sample quality. Exploiting the training data is a crucial step in improving the training efficiency and effectiveness hence the final performance. Concerning intra-sample structure, we exploit the structure information in conversations for context enhancement and self-supervised learning to improve the performance of emotion recognition. As for inter-sample quality, we investigate data uncertainty of large-scale datasets for data rejuvenation and self-training sampling to boost the performance of machine translation. The contributions are summarized as follows:

- For context enhancement, we propose a hierarchical gated recurrent units (HiGRU) to capture both the context of words in utterances and the context of utterances in conversations [39].

To learn long-range context better, we also incorporate a self-attention layer upon the words and utterances, respectively. Experimental results on three ERC datasets show that the HiGRU approach improves the performance significantly, demonstrating the effectiveness of our HiGRU in exploiting the intra-sample structure information.

- For self-supervised learning, we propose a conversation completion task to utilize the sequential relationship of utterances as self-supervised signals for pre-training [40]. Specifically, we pre-train a context-dependent encoder (Pre-CODE) and achieve significant improvements of performance on the ERC tasks after fine-tuning. Extensive analyses show that the pre-trained parameters at both utterance- and conversation-level play a noticeable effect on the final performance, indicating the effectiveness of our pre-training task in exploiting the intra-sample structure information.
- For uncertainty-based data rejuvenation, we divide the large-scale datasets into active data and inactive data by their uncertainty computed based on pre-training MT models [41]. We empirically demonstrate that the existence of inactive data is not model-specific and mainly dependent on the data distribution. Further, we propose data rejuvenation (DataReju) to re-label the inactive data, which is then re-used together with the active data to train the MT models. Experiments on benchmark translation tasks show that our DataReju obtains consistent and significant improvements over strong baselines, demonstrating the effectiveness of exploiting inter-sample quality information.
- For uncertainty-based self-training sampling, we propose the uncertainty-based sampling (UncSamp) strategy to leverage monolingual more efficiently when performing self-training for the MT tasks [42]. Specifically, we compute the translation uncertainty of monolingual sentences based on a bilingual dictionary extracted from the authentic bilingual data, and prefer monolingual sentences with relatively high uncertainty when

sampling. Experiments on the benchmark MT tasks show that the UncSamp approach obtains further improvements over strong self-training baselines, demonstrating the necessity of distinguishing samples.

1.3 Publications During Ph.D. Study

During my Ph.D. study period, we have five research works published at top peer-reviewed conferences, as shown below. Among them, the papers [1,3,4,5][8] correspond to the four contributions introduced in Section 1.2, respectively, which will be elaborated in this thesis.

1. **Wenxiang Jiao**, Haiqin Yang, Irwin King, Michael R. Lyu. "HiGRU: Hierarchical Gated Recurrent Units for Utterance-Level Emotion Recognition". In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), pp. 397-406, Minneapolis, USA, June 2 - June 7, 2019.

2. **Wenxiang Jiao**, Michael R. Lyu, Irwin King. "Real-Time Emotion Recognition via Attention Gated Hierarchical Memory Network". In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020), pp. 8002-8009, New York, USA, February 7 - February 12, 2020.

3. **Wenxiang Jiao**, Michael R. Lyu, Irwin King. "Exploiting Unsupervised Data for Emotion Recognition in Conversations". In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Findings of EMNLP (EMNLP-Findings 2020), pp. 4839-4846, Online, USA, November 16 - November 20, 2020.

4. **Wenxiang Jiao**, Xing Wang, Shilin He, Irwin King, Michael R. Lyu, Zhaopeng Tu. "Data Rejuvenation: Exploiting Inactive

[8] We do not elaborate the 2rd paper since it shares some similar designs of models as the 1st paper but explores a different application scenario.

Training Examples for Neural Machine Translation". In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020), pp. 2255-2266, Online, USA, November 16 - November 20, 2020.

5. **Wenxiang Jiao**, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael R. Lyu, Irwin King. "Self-training Sampling with Monolingual Data Uncertainty for Neural Machine Translation". In Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), To appear, Online, Thailand, August 1 - August 6, 2021.

1.4 Thesis Organization

The remainder of this thesis is organized as follows.

- **Chapter 2**
  In this chapter, we provide a systematic review of the background knowledge and related work. Firstly, we briefly introduce different methods on exploiting intra-sample structure for context modeling in §2.1. Then, §2.2 presents the explorations of self-supervised learning based on intra-sample structure. §2.3 provides the background of data re-weighting, including dynamic weighting, data selection and curriculum learning, which are critical for exploiting inter-sample quality information. Subsequently, in §2.4, we provide some knowledge about data augmentation, which is closely combined with data re-weighting in this thesis. At last, §2.5 provides basic information about the datasets, inference, and evaluation methods involved in this thesis.

- **Chapter 3**
  This chapter presents our investigation on intra-sample structure for context enhancement to perform the ERC tasks. We first introduce the background and our motivation in §3.1 and then

elaborate our proposed approach in §3.2. In §3.3, we conduct experiments to evaluate our approach and compare it with related works. We further provide analyses on model size and case studies in §3.4 to gain a deeper understanding of our approach. Finally, we summarize the work in §3.5.

- **Chapter 4**
  In this chapter, we introduce our study on intra-sample structure for self-supervised learning to leverage unlabeled conversation data. We first introduce the background knowledge in §4.1. Then we elaborate our pre-training tasks and the models in §4.2. In experiments, we conduct experiments for both pre-training in §4.3 and fine-tuning in §4.4, and show the effectiveness of our approach. We further conduct analyses to understand how pre-training improves the performance of the ERC tasks in §4.5. Finally, we conclude the work in §4.6.

- **Chapter 5**
  This chapter presents our study on inter-sample quality for uncertainty-based data rejuvenation. We first introduce the background and motivation in §5.1. Then we elaborate our approach for both identification and rejuvenation of inactive data in §5.2. In §5.3, we identify the inactive data, demonstrate its reasonableness, and present the performance improvements attained by our approach. Further, we conduct extensive analyses to understand the inactive data and the proposed data rejuvenation approach in §5.4. Finally, we summarize the work in §5.5.

- **Chapter 6**
  In this chapter, we show our exploration of inter-sample quality for uncertainty-based self-training sampling. We first introduce the motivation in §6.1 and conduct preliminary experiments to demonstrate the necessity of distinguishing monolingual data in §6.2. We then introduce our solution to the problem in §6.3 and test its effectiveness in §6.4. Finally, we provide analyses

to understand how the proposed approach improves translation quality. At last, we conclude the work in §6.6.

- **Chapter 7**
  In the last chapter, we first summarize the thesis in §7.1. Then in §7.2, we discuss several potential research directions about data exploitation in the future, focusing on data augmentation and self-supervised learning for the low-frequency issue.

□ **End of chapter.**

Chapter 2

Background Review

This chapter reviews the background knowledge and related work. The overall structure is illustrated in Figure 2.1. We first present the background of context modeling in Section 2.1 for sentences and hierarchical texts, the latter of which contain richer intra-sample structure information that will be exploited in this thesis. In Section 2.2, we introduce the representative studies regarding self-supervised learning in different granularity, including word-level, sentence-level, and hierarchical structure, the last of which with rich intra-sample structure information is explored to learn representations from unlabeled conversation data. In Section 2.3, we will introduce data re-weighting, including dynamic weighting, data selection and curriculum learning, which are critical for exploiting inter-sample quality in large-scale datasets. Subsequently, in Section 2.4, we provide some knowledge about data augmentation, which is closely combined with data re-weighting in this thesis. Finally, in Section 2.5, we include the datasets, inference methods, and evaluation metrics that are adopted in this thesis.

2.1 Context Modeling

In this section, we review the background of intra-sample structure for context modeling, including sequential modeling and hierarchical modeling.

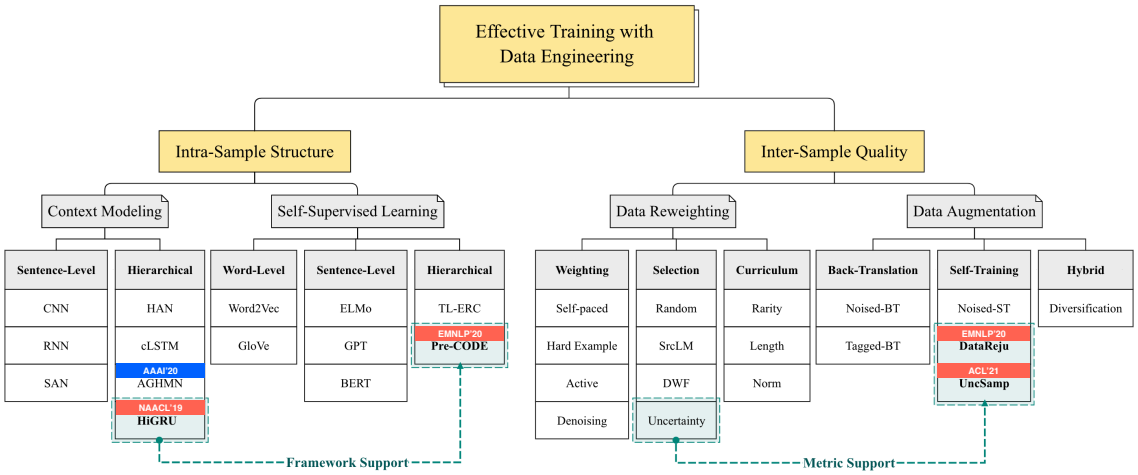


Figure 2.1: Overview of the research on effective training with data engineering. Cells marked with publication venues (e.g., AAAI'20) represent our studies, among which the venues in red are included in this thesis.

2.1.1 Sequential Modeling

In a coarse view, a text sample can be considered as a sequence of tokens (including words and punctuations). Representative neural networks, including the convolutional neural network (CNN), the recurrent neural network (RNN), and the self-attention network (SAN), are all trying to model this sequential relationship. Essentially, modeling the sequential relationship is modeling the context of tokens.

CNN. Convolutional neural networks are basically several convolutions with nonlinear activation functions such as Tanh and ReLU [43]. Though originally invented for computer vision, CNNs have proven to be powerful in performing the text classification task [1, 44, 45]. The model does not need to be a complex one to realize strong results. For example, Kim [1] proposed a very simple CNN with only one layer for text classification but achieved the state-of-the-art results across several benchmark datasets.

Figure 2.2 presents the architecture of the one layer CNN in [1] with two channels for a text sample. Suppose the sentence contains T tokens denoted by indexes in the vocabulary $W = w_1, w_2, \cdots, w_T$, and each word corresponds to a vector with a size of d_w in the word

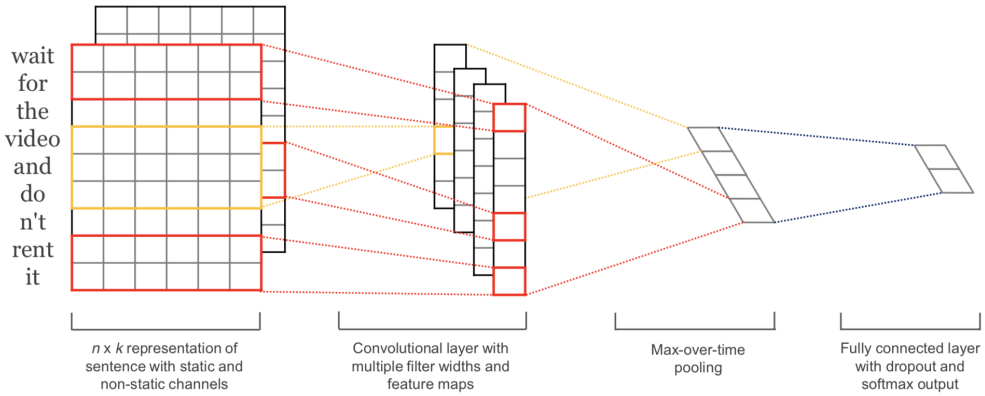


Figure 2.2: The architecture of one layer CNN with two channels for a text sample [1].

embedding matrix. Thus, the text sample is represented as:

$$\mathbf{x}_{1:T} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \cdots \oplus \mathbf{x}_T, \tag{2.1}$$

where $\mathbf{x}_i \in \mathbb{R}^{d_w}, i \in [1, T]$, and $\oplus$ denotes concatenation. With a filter size of k, CNN can extract a feature c_i from a window of words $\mathbf{x}_{i:i+h-1}$ as below:

$$c_i = f(\mathbf{W} \cdot \mathbf{x}_{i:i+k-1} + b). \tag{2.2}$$

Here, $\mathbf{W} \in \mathbb{R}^{kd_w}$ is the weight, $b \in \mathbb{R}$ is the bias term, and f is an activation function. The filter goes through the word sequence and produces a feature map as:

$$\mathbf{c} = [c_1, c_2, \cdots, c_{T-k+1}], \tag{2.3}$$

with $\mathbf{c} \in \mathbb{R}^{T-k+1}$.

Subsequently, a max-over-time pooling operation is applied over the feature map, and the maximum value $\hat{c} = max\{\mathbf{c}\}$ is taken as the feature of this filter. In practice, a CNN model involves multiple filters (e.g., $k = 3, 4, 5$) and a number of feature maps (e.g., n feature maps hence $\mathbf{W} \in \mathbb{R}^{kd_w \times n}$). The produced features are concatenated as the representation of the sentence. While CNNs are usually used for text classification, they have also been adapted for sequence-to-sequence generation [14] to enable parallel computation, which is infeasible for RNN-based encoder-decoder models due to the recurrence.

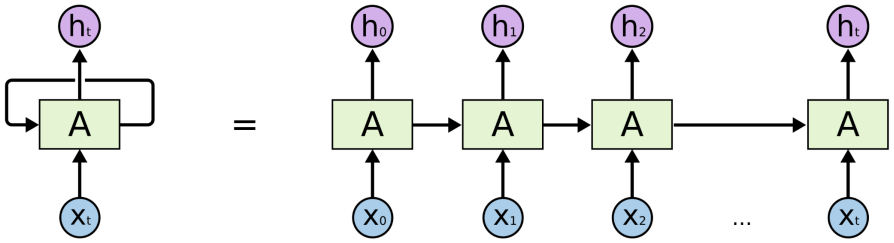


Figure 2.3: The unfold recurrent neural network [2].

RNN. Recurrent neural network is a kind of architecture that enables the persistence of previous events. As presented in Figure 2.3, the output of RNN is conditioned on the current input and the output of last time-step[1]. Therefore, RNN is particularly suitable for tasks involving sequential inputs, such as language modeling, reading comprehension, and machine translation. However, standard RNNs are incapable of long-term dependencies. To address this problem, Hochreiter and Schmidhuber [46] introduced the Long Short-Term Memory networks (LSTMs), which have been refined and popularized in recent decades, for example, the Gated Recurrent Unit (GRU) [47].

Formally, an **LSTM** is composed of a forget gate, an input gate, a new state, a cell state, and an output state. The forget gate $f_t \in \mathbb{R}^{d_1}$, made by a sigmoid layer, decides what information to be thrown away from the cell state $c_t \in \mathbb{R}^{d_1}$. The input gate $i_t \in \mathbb{R}^{d_1}$ decides which values of the new state $\tilde{c}_t \in \mathbb{R}^{d_1}$ to be updated in the cell state. The output gate decides what to output based on the cell state. The formulations are as below:

$$f_t = \sigma(V_f \cdot x_t + W_f \cdot h_{t-1}), \tag{2.4}$$
$$i_t = \sigma(V_i \cdot x_t + W_i \cdot h_{t-1}), \tag{2.5}$$
$$\tilde{c}_t = \tanh(V_c \cdot x_t + W_c \cdot h_{t-1}), \tag{2.6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{2.7}$$
$$o_t = \sigma(V_o \cdot x_t + W_o \cdot h_{t-1}), \tag{2.8}$$
$$h_t = o_t \cdot \tanh(c_t), \tag{2.9}$$

where $x_t \in \mathbb{R}^{d_0}$ is the input vector, $V, W \in \mathbb{R}^{d_1 \times d_0}$ are weight

[1] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

matrices, $\odot$ denotes element-wise multiplication, and d_0 and d_1 are the dimensions of inputs and hidden states, respectively.

As for **GRU**, the essence of it is a gating mechanism of RNN yielding similar performance to LSTM, but consuming lower computational cost [47]. In the t-th step, GRU utilizes the reset gate $r_t \in \mathbb{R}^{d_1}$ and the update gate $z_t \in \mathbb{R}^{d_1}$ to control the information passed from the current input $x_t \in \mathbb{R}^{d_0}$ and the previous hidden state $h_{t-1} \in \mathbb{R}^{d_1}$. The hidden state h_t at the current step is then updated by:

$$z_t = \sigma(V_z \cdot x_t + W_z \cdot h_{t-1}), \tag{2.10}$$

$$r_t = \sigma(V_r \cdot x_t + W_r \cdot h_{t-1}), \tag{2.11}$$

$$\tilde{h}_t = \tanh(V_s \cdot x_t + W_s \cdot (h_{t-1} \odot r_t)), \tag{2.12}$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}, \tag{2.13}$$

where $V, W \in \mathbb{R}^{d_1 \times d_0}$ are weight matrices, d_0 and d_1 are the dimensions of inputs and hidden states, respectively.

In practice, a uni-directional RNN is adopted for language modeling while a bi-directional RNN [12] can obtain more comprehensive representations. These two kinds of RNN can also form an encoder-decoder model [13] to perform sequence-to-sequence generation.

SAN. Before we introduce the self-attention network, we first take a look at the general attention mechanism, the introduction of which is a milestone for sequence-to-sequence generation tasks. The attention network computes the relevance of each value vector based on queries and keys. Intuitively, we can consider the query as what kind of information we are looking for, the key as the relevance to the query, and the value as the actual contents of the input.

Formally, given a set of m query vectors $\mathbf{Q} \in \mathcal{R}^{m \times d}$, a set of n key vectors $\mathbf{K} \in \mathcal{R}^{n \times d}$, and associated value vectors $\mathbf{V} \in \mathcal{R}^{n \times d}$, the computation of attention network involves two steps. The first step is to compute the relevance between queries and keys:

$$\mathbf{R} = \text{score}(\mathbf{Q}, \mathbf{K}), \tag{2.14}$$

where score$(\cdot)$ is the score function, and $\mathbf{R} \in \mathcal{R}^{m \times n}$ stores the relevance

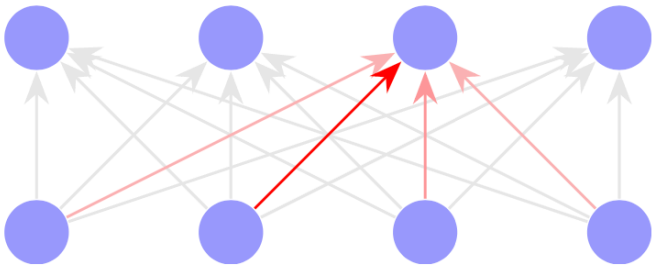


Figure 2.4: The architecture of SAN [3].

score between each key and value. The second step is to compute the output vector. For each query vector, the corresponding output vector is a weighted sum of value vectors:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{R}) \cdot \mathbf{V}. \tag{2.15}$$

Considering the choice of score function, the attention network can be categorized into two classes, i.e., additive attention and dotproduct attention. The additive attention compute scores through a feed-forward neural network:

$$\mathbf{R}_{[i,j]} = \mathbf{v}^\top \tanh(\mathbf{W}_s \mathbf{Q}_{[i]} + \mathbf{U}_s \mathbf{K}_{[j]}), \tag{2.16}$$

where $\mathbf{W}_s, \mathbf{U}_s \in \mathcal{R}^{d \times d}$, and $\mathbf{v}$ are learnable parameters. As for dotproduct attention, the scores are computed by the dotproduct between each key and value:

$$\mathbf{R}_{[i,j]} = \mathbf{Q}_{[i]}^\top \mathbf{K}_{[j]}. \tag{2.17}$$

Self-attention network [15] is a special expression of the general attention network, in which the query, key and value vectors are the same. Given a sentence with n tokens and the token embeddings $\mathbf{E} \in \mathcal{R}^{n \times d}$, the output by a self-attention layer is expressed as:

$$\text{SAN}(\mathbf{E}, \mathbf{E}, \mathbf{E}) = \text{softmax}(\mathbf{E}\mathbf{E}^\top) \cdot \mathbf{E}. \tag{2.18}$$

Compared to RNN and CNN, SAN enables both the parallel computation and the capturing of long-range context, because the computation of each time-step does not depend on previous time-steps and each time-step can assess its relevance to distant time-steps through

attention. These advantages enables the training of large models on large-scale datasets, making SAN the standard network in existing NLP research.

2.1.2 Hierarchical Modeling

The text sample can also be decomposed into multiple levels from a finer-grained view, especially when it contains several sentences, for example, a document or a conversation. Essentially, the document or the conversation follows a words-to-sentence and sentences-to-document hierarchy. Representative models that consider such a hierarchy include hierarchical attention network (HAN), contextual LSTM (cLSTM), conversational memory network (CMN), and DialogueRNN. In this way, both the context of words and that of sentences are required to be well learned.

HAN. Hierarchical attention network tries to incorporate the knowledge of text structure in the model architecture. Since a text is composed of sentences, which can be further decomposed into words, Yang et al. [4] likewise proposed to extract the representation of a text by first learning representations of sentences and then combining them by attentive pooling. The main difference that HAN holds from other hierarchical networks is the way to extract the higher-level representations from the lower-level ones. Different from CNN-GRNN and LSTM-GRNN [48] that apply max-pooling or mean-pooling, HAN proposes to use attentive pooling because of the observation that different words and sentences in a text are differentially informative.

We take the procedure of extracting text representations for example. Suppose we have a text sample with several sentences, the sentence representations are extracted by some sentence encoder. We can utilize a bidirectional GRU (Bi-GRU) to read the sequence of sentences and produce the contextual sentence representations by $h_l = [\overrightarrow{h}_l, \overleftarrow{h}_l],\ l \in [1, L]$, where L is the number of sentences, and $\overrightarrow{h}_l$ and $\overleftarrow{h}_l$ are the hidden states in the forward and backward direction, respectively.

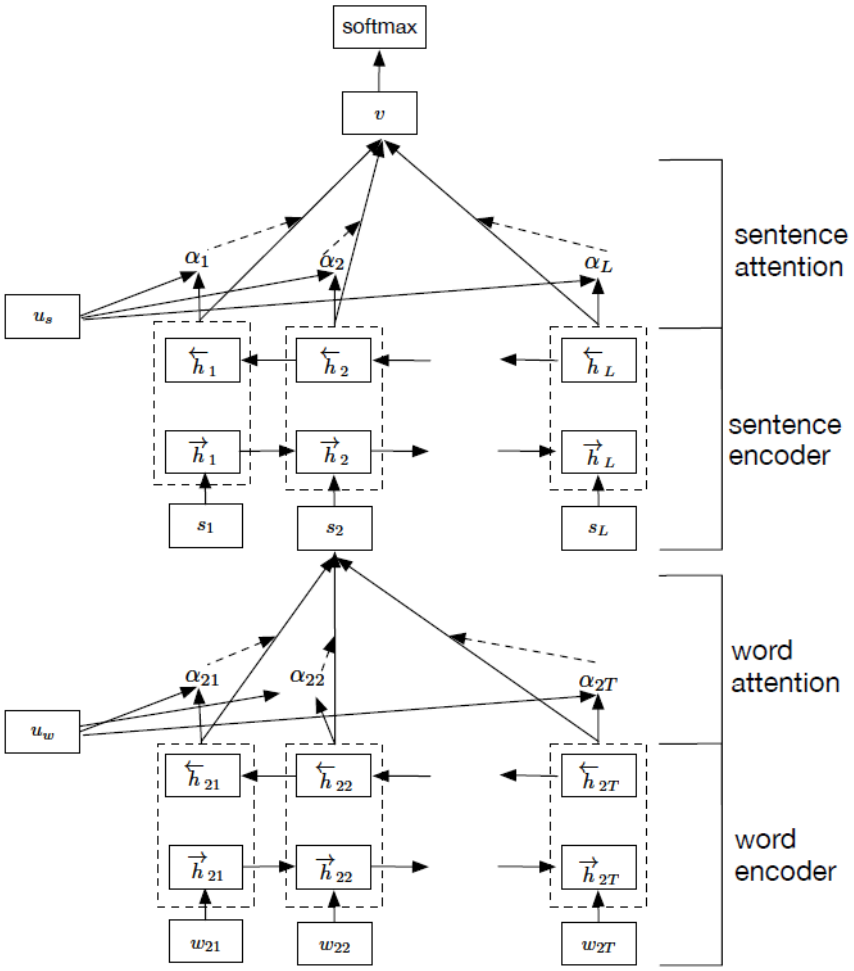


Figure 2.5: The architecture of HAN [4].

The attentive pooling attempts to reward sentences that are clues to classify a text correctly. It introduces a sentence level context vector u_s to measure the importance of the sentences. The formulations are as below:

$$u_l = \tanh(W_s \cdot h_l + b_s), \tag{2.19}$$

$$\alpha_l = \frac{\exp\left(u_l^\top u_s\right)}{\sum_l \exp\left(u_l^\top u_s\right)}, \tag{2.20}$$

$$v = \sum_l \alpha_l h_l, \tag{2.21}$$

where W_s is the weight matrix, b_s is the bias term, α_l is the attention score, and v is the representation of the text. Here, the vector u_s acts as the query, and the projected hidden states u_l serves as both the key and the value in the attentive pooling.

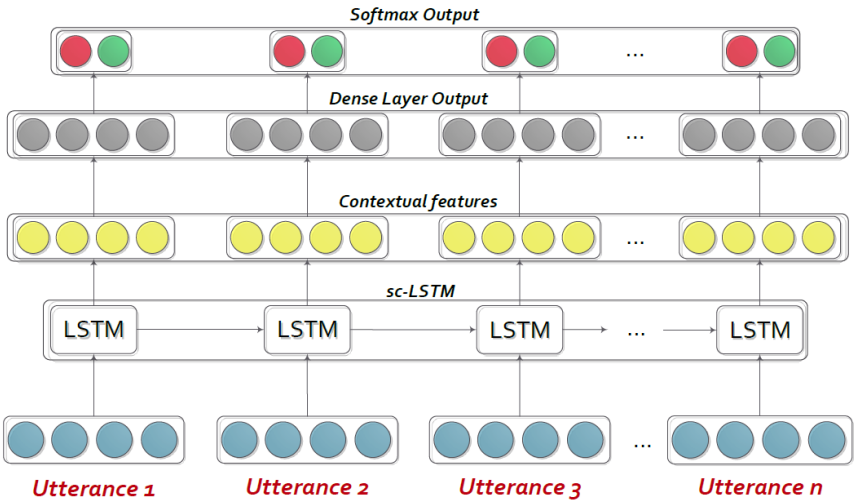


Figure 2.6: The architecture of cLSTM [5].

cLSTM. The motivation of contextual LSTM is based on the fact that the classification of each member in a sequence is dependent on the other members. Since the utterances in a conversation form a sequence, when classifying an utterance, other utterances should provide relevant contextual information. However, previous approaches treat the utterances independently and ignore the order of utterances in conversations. To address the problem, Poria et al. [5] propose an LSTM network that takes the sequence of utterances in a conversation in order to extract contextual features by modeling the dependencies among the input utterances.

The work considers multimodal features, including text, audio, and visual. Before fed to the LSTM, each kind of unimodal feature is extracted for each utterance independently. A two-layer CNN is applied over the input word embeddings to extract the textual features. For the audio feature extraction, an open-source software named openSMILE [49] is adopted to automatically extract audio features such as pitch and voice intensity. As for the visual feature extraction, a 3D-CNN [50] is applied over the video to learn relevant features from each frame and the changes among consecutive frames.

To capture the flow of information triggers across utterances, an LSTM network is adopted to fuse the independent features. As

shown in Fig. 2.6, a simple unidirectional LSTM takes as input the independent features, and this variant is termed as scLSTM. In fact, the work investigates a few variants and finds that a bidirectional LSTM (this variant is termed bcLSTM) performs the best because an utterance can obtain information from other utterances both before and after it.

Generally, cLSTM is suitable for sequence classification in both monologues and dialogues. For different variants, the scenarios could be different, for example, scLSTM better suits real-time sentiment analysis. Because bcLSTM lets each utterance see both the history and feature, which requires all utterances to be inputted at the same time. In contrast, scLSTM only uses a unidirectional LSTM so that the utterances can be inputted one by one. The common limitations for all the variants of cLSTM include: 1) First, the model is not an end-to-end version, since the independent features are extracted beforehand and will not be updated during training. 2) Second, the encoder for textual feature extraction is CNN, which is incompatible with the LSTM over it and could be improved by a bidirectional LSTM or a bidirectional GRU. 3) Third, cLSTM is incapable of long-range summarization and weighted influence from the context.

CMN. Conversational memory network [6] is dedicated to emotion recognition in dyadic conversations. The work argues that emotional dynamics in a conversation are driven by both self- and inter-speaker emotion influence. Self-influence refers to the degree to which a speaker's feelings carry over from one moment to another. Inter-speaker influence means that speakers can affect and be affected by their counterparts in terms of emotional state. Besides, it points out that cLSTM is incapable of long-range summarization and unweighted influence from the context. CMN is efficient in capturing long-range context due to memory nets, which can also decide the importance of each utterance in the context. To model the self and inter-speaker influence, the memory of each speaker is obtained by a GRU each, and a current utterance can attend to both memories.

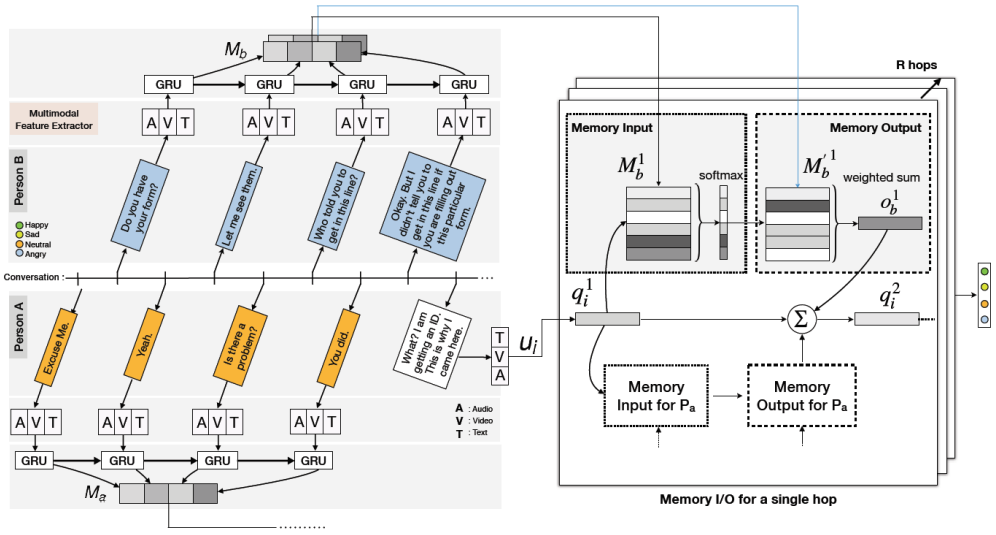


Figure 2.7: The architecture of CMN [6].

The memory network in CMN is inspired by [51]. For each speaker, there are two memory banks generated by two GRUs over the most recent K utterances, respectively. The first memory bank is used to compute the relevance of each memory context with the query (the current utterance) by attention, and the second memory bank is weighted by the attention scores and summed as the memory output. The final representation for each utterance is the concatenation of the query and the memory outputs of each speaker. For multi-hop memory networks, each speaker takes the output memory of the last hop as the input memory of the current hop, and utilizes a new GRU to produce the output memory of the current hop. At every hop, the representation of the query utterance is updated as the sum of the query and the memory outputs at the last hop.

The limitations of CMN include: 1) First, the idea of separating the memory of each speaker is intuitive. However, in practice, there are conversations involving many speakers, resulting in the complication of modeling and the data sparsity of each speaker. 2) Second, the use of multi-hop memory networks induces a linear increase of the model parameters with respect to the number of hops. 3) The representation of the query utterance does not take the history into account. 4) The memory bank is generated by unidirectional GRUs, preventing each

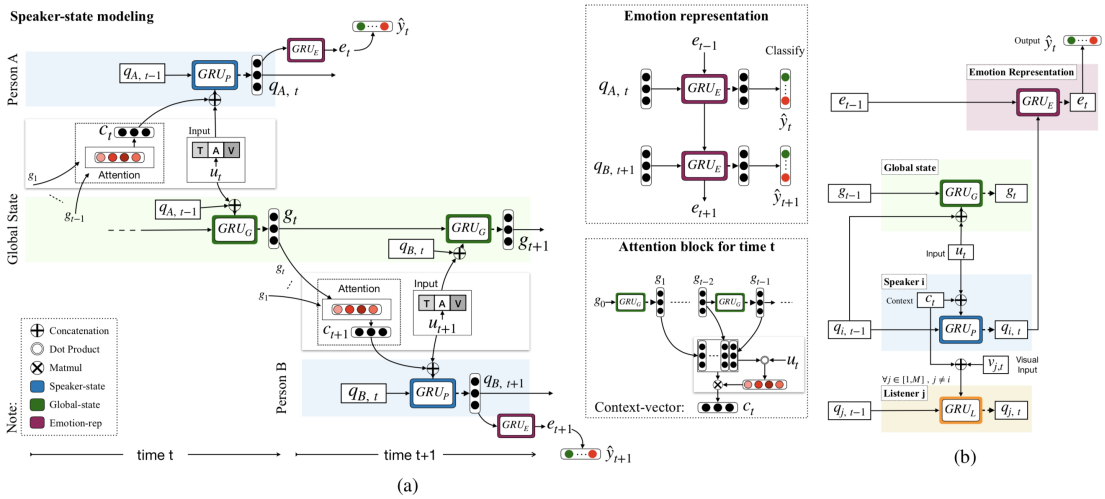


Figure 2.8: The architecture of DialogueRNN [7].

utterance from seeing the utterances after it.

DialogueRNN. The DialogueRNN [7] claims that the prediction of emotion is affected by three main factors, namely, the speaker, the context given by the preceding utterances, and the emotion behind the preceding utterances. Firstly, to track the state of speakers throughout the conversation, each speaker is assigned with a *party state* which is updated by a GRU once the speaker utters an utterance. Secondly, to model the context of an utterance, DialogueRNN utilizes a *global state* which is shared among the parties. The *global state* is updated by jointly encoding the preceding utterances and the party states. Thirdly, DialogueRNN models the evolution of emotion by updating the emotion representation based on the speaker's state and the emotion representation of the last utterance.

Regarding our studies on hierarchical context modeling, we proposed to fully utilize the intra-sample structure information in the conversations, for example, the bidirectional relationship between words or utterances, and the long-range context, which are not well captured in previous works. Based on these structure information, we developed the attention gated hierarchical memory network (AGHMN) [52] and the hierarchical gated recurrent units (HiGRU) [39], the latter of which will be elaborated in this thesis. Due to the small scale of datasets,

early studies on the ERC tasks develop the models mainly based on RNNs (e.g., LSTM, and GRU). However, researchers also try to adopt the Transformer models to either directly model the task [53] or extract features from the pre-trained Transformer models [54], as Transformer models are effective in capturing long-range context [15, 55].

2.2 Self-Supervised Learning

In this section, we review the background of intra-sample structure for self-supervised representation learning by pre-training. Pre-training is a learning paradigm to train the model on other tasks with large-scale datasets before starting the target task. Usually, pre-training is conducted in a self-supervised fashion on large-scale unlabeled datasets. Specifically, we introduce word-level pre-training, sentence-level pre-training, and hierarchical pre-training, which learn the word representations by considering the discrete words, the entire sentence, and the whole document or conversation, respectively.

2.2.1 Word-Level Pre-Training

Word2Vec. For decades, the n-gram based models have been dominating the language modeling field, due to their simplicity and low complexity of computation. With the progress of machine learning in recent years, it becomes possible to train more complex models on much larger datasets. For example, language models based on neural network learning significantly outperform n-gram models [56, 57, 58]. But these architectures are facing high computation costs between the projection and the hidden layer, because the values in the projection layer are dense.

To reduce the computation complexity, Mikolov et al. [59] proposed two shallow neural network architectures, i.e., the skip-gram model and the continuous bag-of-words model. In the meantime, to handle the intractability of full softmax function at the output, several solutions were proposed, either using hierarchical versions of softmax [60, 22] or unnormalized models for training [61]. Among

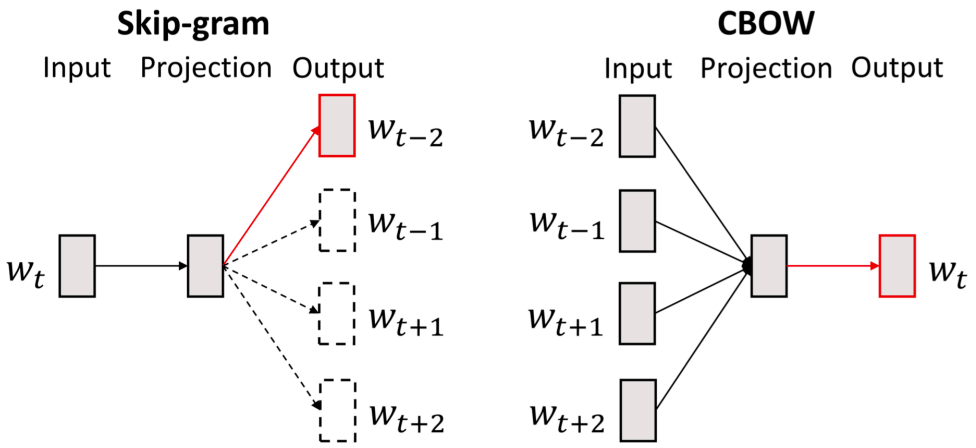


Figure 2.9: The architectures of Skip-gram and Continuous Bag of Words models.

these variants of the skip-gram model, the skip-gram model with negative sampling [22] has achieved state-of-the-art results across several evaluation tasks of word embeddings, namely the analogy reasoning, sentiment analysis, sentence completion, and so on.

These models use a shallow neural network with only one hidden layer to learn the relationship between each word and its context words and obtains the hidden weights as word vectors. It is capable of learning semantic and syntactic meanings of words, and mapping similar words into nearby locations in the vector space. The simplicity enables it to train on huge datasets with billions of tokens within a short time. By arithmetic operations on word vectors, it is able to produce meaningful phrases, which is quite amazing.

GloVe. Right after Word2Vec, GloVe [23] combines the global matrix factorization methods (e.g., LSA) and the local context window methods (e.g., Word2Vec) to improve the quality of word representations. While methods like LSA leverage statistical information effectively, they do relatively poorly on the word analogy task, implying a sub-optimal vector space structure. Methods like Word2Vec learn word vectors by training on separate local contexts which do not make full use of the global statistics of the corpus. In this sense, GloVe is kind of like finding a learning algorithm on the global statistics. In fact, that is exactly what the authors have done. They proposed a specific weighted least squares model that was trained on global word-word

co-occurrence counts and thus made efficient use of statistics.

The weighted least squares model of GloVe [23] is expressed as:

$$J = \sum_{i,j=1}^{|V_w|} f(X_{ij})(v_{w_i} \top v'_{w_j} + b_i + b'_j - \log X_{ij})^2, \tag{2.22}$$

where $|V_w|$ is the size of vocabulary, v_{w_i} and v'_{w_j} are the vector representations of a word and its context, and b_i and b'_j are the corresponding bias. $f(X_{ij})$ is a weighting function over the co-occurrence matrix X_{ij}. It is proposed for fixing the problem that very frequent co-occurrences are actually not that relevant and may cause extra noises. The weighting function adopted in GloVe [23] is expressed as:

$$f(x) = \begin{cases} (x/x_{max})^\alpha & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}, \tag{2.23}$$

where x_{max} is the threshold set as 100, and α is a hyperparameter said to work well with the value of 3/4. It is interesting that a similar distortion value for negative sampling was found to give the best performance in Word2Vec [22].

As for the construction of co-occurrences matrix X_{ij}, there are several decisions waited to be decided. We need to choose how large the context window should be and whether to distinguish the left context from the right context. In the paper, the authors use a decreasing weighting function so that word pairs that are d words apart contribute $1/d$ to the total count. This is a way to account for the fact that close contexts are more relevant to the word than those very distant ones.

2.2.2 Sentence-Level Pre-Training

ELMo. While word-level representation learning improves the performance of NLP tasks significantly, it performs badly on polysemy as it does not consider the context of the whole sentence. To solve such a problem, Peters et al. [24] proposed to learn deeply contextualized

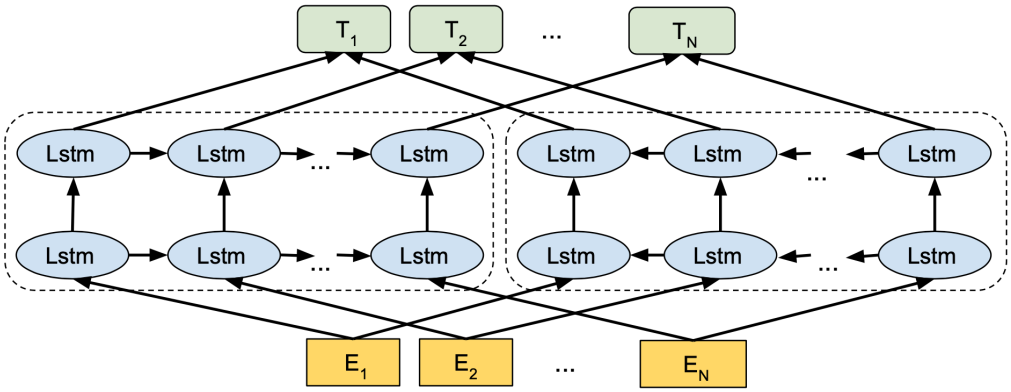


Figure 2.10: The architecture of ELMo [8].

word embeddings from language models, called ELMo, which are pre-trained on large-scale corpora. ELMo is supposed to learn both words (e.g., syntax and semantics) and linguistic context.

Given a sentence with n tokens, $(x_1, x_2, \ldots, x_n)$, a forward language model computes the probability of the sentence by modeling the probability of each token x_i conditioned on its history $(x_1, x_2, \ldots, x_{i-1})$:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots . x_{i-1}). \tag{2.24}$$

Correspondingly, the sentence can also be modeled by a backward language model, which computes the probability of each token by the tokens after it:

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{i+1}, x_{i+2}, \ldots . x_n). \tag{2.25}$$

As shown in Figure 2.10, ELMo combines both the forward and backward language models, each modeled by a LSTM network, and trains the model by maximizing the negative log-likelihood:

$$\mathcal{L} = -\sum_{i=1}^{n} \Big(\log p(x_i | x_1, x_2, \ldots . x_{i-1}; \overrightarrow{\Theta}) \tag{2.26}$$

$$+ \log p(x_i | x_{i+1}, x_{i+2}, \ldots, x_n; \overleftarrow{\Theta}). \tag{2.27}$$

For each token x_i, an L-layer ELMo computes a set of $2L + 1$ representations, two for each layer and one for the embedding layer.

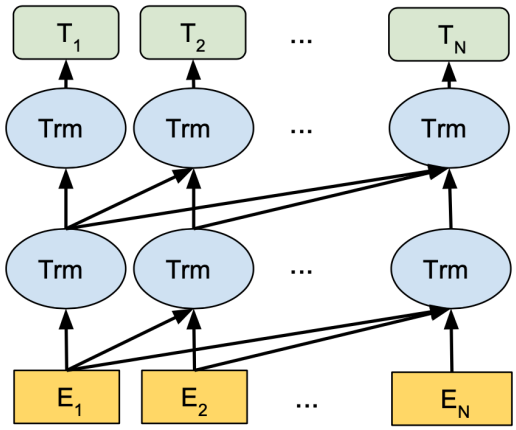


Figure 2.11: The architecture of GPT [8].

When transferring the pre-trained EMLo for downstream tasks, the $2L + 1$ representations are summarized with weights, which are learnable parameters in the fine-tuning stage.

GPT. Due to the stronger performance of Transformer models [15], researchers have been developing deep contextualized word embeddings by Transformer-based language models. GPT [26] is the first and representative one, performing generative pre-training on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task. Unlike the original transformer architecture, GPT discards the encoder part and only uses the decoder. Thus, there is only one single input sentence rather than two separate source and target sequences. Each transformer block contains a masked multi-headed self-attention followed by a pointwise feed-forward layer and normalization layers in between. The final output produces a distribution over target tokens after softmax. The computation of probability for each token is the same as ELMo but without the backward computation.

There are two main differences between GPT and ELMo: 1) First, ELMo uses the concatenation of forward and backward LSTM networks while GPT only adopts a multi-layer transformer decoder. 2) For downstream tasks, ELMo uses the unsupervised feature-based approach, while GPT fine-tunes the same pre-trained model. So far,

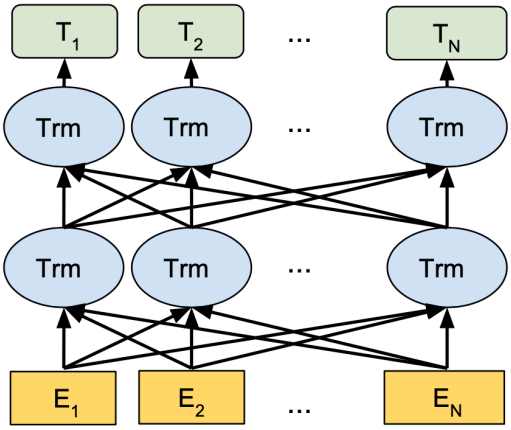


Figure 2.12: The architecture of BERT [8].

GPT has been promoted to more powerful versions (i.e., GPT2 and GPT3) [27, 28] by increasing the model capacity and the data scale.

BERT. As introduced above, GPT is actually a causal language model that reads a sentence in the forward direction. Devlin et al. [8] argue that the language model only captures the context of words before them without that after them, and propose a masked language model to learn better word representations, i.e., the so-called BERT. With the masked language model, BERT computes the representation of each word from the unlabeled text by jointly conditioning on both its left and right context in all layers. This is also the largest difference of BERT from GPT. In addition to the masked language model tasks, BERT is also trained on the next sentence prediction task to capture the relationship between sequences.

For the masked language model task, BERT chooses 15% of the token positions from the text corpus at random for prediction. If the i^{th} token is chosen, we replace the i^{th} token with: 1) the `[MASK]` token for 80% of the time; 2) a random token for 10% of the time; and 3) the unchanged i^{th} token for the rest 10% of the time. The next sentence prediction task is designed for downstream tasks like question answering and natural language inference, which require the understanding of the relationship between two text sentences, which cannot be directly captured by language modeling. Specifically, BERT

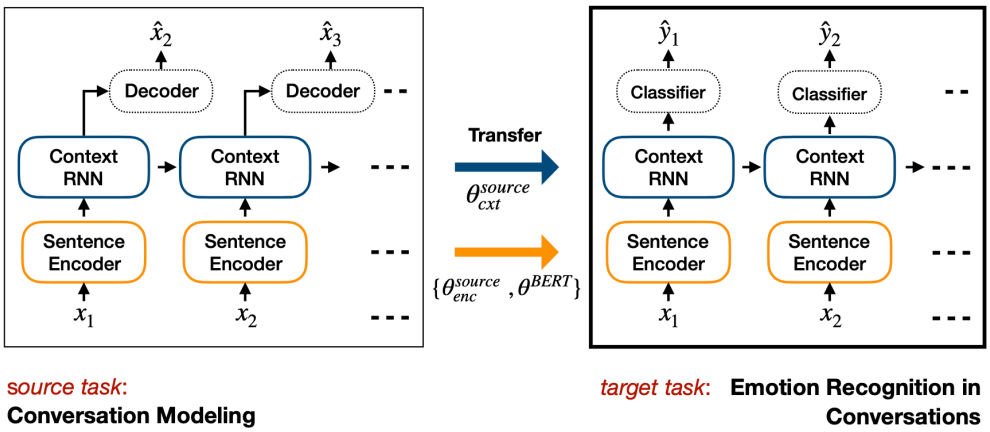


Figure 2.13: The architecture of TL-ERC [9].

is trained as a binary classifier to tell whether one sentence is the next sentence of the other. Specifically, when choosing the sentences `A` and `B` for each pre-training sample, 50% of the time `B` is the actual next sentence that follows `A`, and 50% of the time it is a random sentence from the corpus.

In order to transfer the pre-trained model for downstream tasks directly, BERT is designed to prepend a special token `[CLS]` for each input sentence. The hidden state of the `[CLS]` token is usually used as the representation of the whole input sentence.

2.2.3 Hierarchical Pre-Training

TL-ERC. While most advances of self-supervised representation learning are achieved at sentence-level, there also appear attempts for more complex text formats. For example, TL-ERC [9] pre-trains a hierarchical recurrent encoder-decoder model on conversations to learn representations for the ERC tasks. In specific, the model consists of three sequential components: 1) The sentence encoder RNN for the encoding of utterances; 2) The context encoder RNN for modeling the conversational context of utterances; and 3) The decoder RNN for generating the response utterance. For the downstream ERC task, the sentence encoder and the context encoder can be directly transferred, while the decoder is replaced with a discriminative mapping to the label space of emotions.

As the concurrent studies, the main difference between TL-ERC and our PRE-CODE [40] is the pre-training objective: TL-ERC models the conversation in a response generation task while our PRE-CODE in a contrastive response retrieval task, which is a more light-weight and effective method. More recent studies also try to pre-train large models on unlabeled conversation data by the encoder-decoder model [62] or a tree-based attention network [63], which indicating the potential of self-supervised learning on conversations.

2.3 Data Re-Weighting

In this section, we review the background knowledge of inter-sample quality exploitation with data re-weighting, including dynamic weighting, data selection, and curriculum learning.

2.3.1 Dynamic Weighting

Importance Weighting. In machine learning, there are plenty of studies trying to ease the training or accelerate the convergence of models by distinguishing training samples. They assign different importance to the training samples to re-weight the loss function. According to the choices of preferred samples, these studies can be divided into three categories, namely, self-paced learning, hard sample mining, and active learning. Self-paced learning [19] prefers the easy samples during training while hard samples mining [20] proposes to pay more attention to the hard samples, which are supposed to be more informative. Active learning [21] makes a compromise between self-paced learning and hard sample mining by emphasizing the high-variance samples. The difficulty of each sample can be calculated by the model confidence and its variance across a number of training steps. During training, higher weights can be assigned to the loss function of the preferred samples.

Data Denoising. Data denoising [64] is concerned with the quality of training data and tries to reduce the negative effect of data noise on NMT models. The noise in a sentence pair is defined in terms of the

Algorithm 1: Denoising NMT training with trusted data and online data selection [64].

Input: Noisy data $\tilde{\mathcal{D}}$, trusted data $\hat{\mathcal{D}}$
Output: A denoised, better model
$t = 0$; Randomly initialize $\tilde{\Theta}_0$.
while $t < T$ **do**
  $p(\mathbf{y}|\mathbf{x}, \hat{\Theta}_t) \leftarrow \text{denoise}(\text{p}(\mathbf{y}|\mathbf{x}, \tilde{\Theta}_\text{t}), \hat{\mathcal{D}})$.
  Randomly draw $\tilde{\mathcal{B}}_t^{\text{random}}$ from $\tilde{\mathcal{D}}$.
  Compute the noise score for samples in $\tilde{\mathcal{B}}_t^{\text{random}}$.
  Sort $\tilde{\mathcal{B}}_t^{\text{random}}$ by the noise scores.
  Sample b_t from top r_t of above sorted buffer.
  Train $p(\mathbf{y}|\mathbf{x}, \tilde{\Theta}_t)$ on b_t to produce a new model $p(\mathbf{y}|\mathbf{x}, \tilde{\Theta}_{t+1})$.
  Discard the denoised model $p(\mathbf{y}|\mathbf{x}, \hat{\Theta}_t)$.
  $t \leftarrow t + 1$.
end

comparison between a noisy model $\tilde{\Theta}$ and a denoised model $\hat{\Theta}$. The noisy model is trained on large scale noisy datasets. For the denoised model, we can fine-tune the noisy model on a set of trusted data, i.e., clean in-domain data. Usually, the clean in-domain data contains limited samples, so fine-tuning with a small batch and a small learning rate is required to prevent over-fitting. Therefore, the denoised model is supposed to be a more accurate probability distribution than the noisy model. Then, taking NMT as an example, we can compute the noise score of a sentence pair $(\mathbf{x}, \mathbf{y})$ as:

$$\text{noise}(\mathbf{x}, \mathbf{y}; \tilde{\Theta}, \hat{\Theta}) = \log p(\mathbf{y}|\mathbf{x}, \tilde{\Theta}) - \log p(\mathbf{y}|\mathbf{x}, \hat{\Theta}). \tag{2.28}$$

The noise score can be normalized by the length of the target sentence. The bigger the $\text{noise}(\mathbf{x}, \mathbf{y}; \tilde{\Theta}, \hat{\Theta})$ is, the higher the noise level of the sentence pair shows.

During the training of the NMT models, we will select the sentence pairs with the lowest noise score to train the model. The noisy model can be updated dynamically on the selected data, which will be fine-tuned on the trusted data for a new denoised model. In this way, the NMT model can be trained with the data noise removed gradually, resulting in better translation performance. Algorithm 1 shows the details of the process.

2.3.2 Data Selection

Data selection can be regarded as the hard version of data re-weighting. Both dynamic weighting and data selection emphasize the preferred samples, except that dynamic weighting differentiates the importance of samples during training while data selection directly discards the samples that are not favored before training.

Language Model Selection. Language models are usually adopted for in-domain data filtering, which can be performed by an in-domain language model [30] or together with an out-of-domain (OOD) language model. Given an in-domain corpus D_I and an OOD corpus D_O, we would like to find a subcorpus $D_{O,Sub}$ from D_O that is drawn from the same distribution as D_I. For any s, using Bayes' rule, we can calculate the probability a sentence s in D_O is drawn from $D_{O,Sub}$:

$$P(D_{O,Sub}|s, D_O) = \frac{P(s|D_{O,Sub})P(D_{O,Sub}|D_O)}{P(s|D_O)}. \tag{2.29}$$

We can ignore the $P(D_{O,Sub}|D_O)$ term since it will be constant for any given D_I and D_O, and use $P(s|D_I)$ instead of $P(s|D_{O,Sub})$ as they are drawn from the same distribution. Moving into log domain, we can calculate the probability score for the sentence s as $\log P(D_{O,Sub}|s, D_O) \propto \log P(s|D_I) - \log P(s|D_O)$ where $P(s|D_I)$ and $P(s|D_O)$ can be estimated by language models LM_I and LM_O trained on D_I and D_O, respectively. While neural language models perform better than n-gram models [65], the latter is still favored when the corpus is very large due to the efficiency of training and evaluation.

Difficult Word Selection. Selecting in-domain data may enhance the learning of originally frequent words; however, researchers also attempt to take advantage of OOD words to increase the training signal. For example, Fadaee et al. [31] find that in back-translation, words with high prediction loss during training benefit most from the addition of synthetic data. Therefore, they propose to sample monolingual data with difficult-to-predict words using prediction loss and word frequency (eg., low-frequency words). We can rely on a word rarity metric [10] to score each monolingual sentence and select the

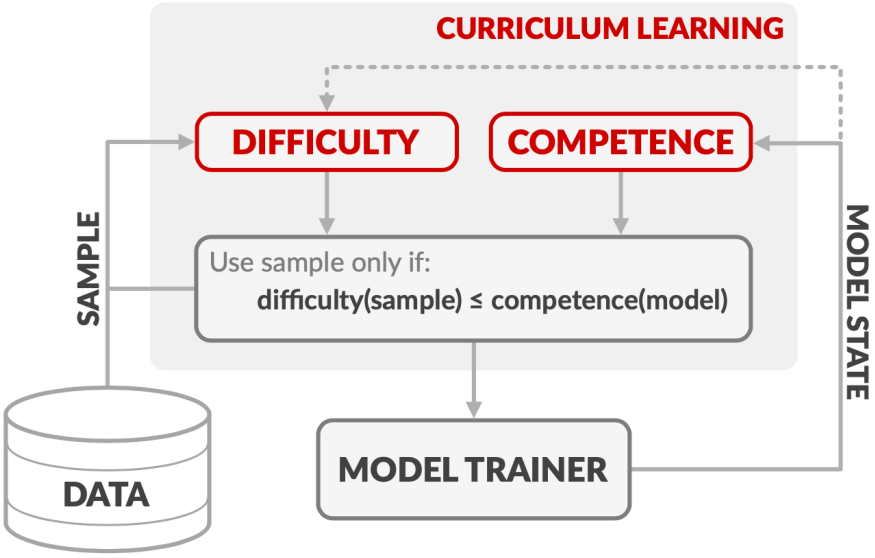


Figure 2.14: Overview of the competence-based curriculum learning [10].

desired sentences.

Data Redundancy. Birodkar et al. [66] reveal that data redundancy exists in large-scale image recognition datasets, e.g., CIFAR-10 [67] and ImageNet [68] datasets. They find that a subset can generalize on par with the full dataset and that at least 10% of training data are redundant in these large-scale image classification datasets.

2.3.3 Curriculum Learning

Instead of dynamically weighting the training samples, curriculum learning points out that the order of feeding training samples also affect the training process significantly. Usually, curriculum learning suggests to feed easy samples into the models firstly and gradually increase the difficulty of samples, so that the models can learn faster at the beginning of training. Curriculum learning has been successfully applied to the training of NMT models recently [10, 18, 33, 34, 35, 69].

As the name suggests, curriculum learning defines a curriculum for models to learn from training data. Generally, based on the estimated difficulty of a sample and the current competence of the model, curriculum learning decides which training samples are shown to the model at different times during training, usually from easy

to hard. The idea originates from the fact that human learns easy knowledge first before moving to hard knowledge. The metrics for determining the difficulty of a sample could be derived from some linguistic properties [10] (e.g., sentence length and word rarity), data uncertainty, or embedding norm [18]. Curriculum learning is demonstrated to have the following advantages, namely, reducing training time and the need for specialized training tricks, and improving the overall performance.

Regarding our studies on data re-weighting, we mainly rely on the metrics from data selection to identify inactive samples or uncertain samples from large-scale datasets. As a result, we have proposed the data rejuvenation (DataReju) framework [41] and the uncertainty-based self-training sampling (UncSamp) strategy to boost the performance of large-scale NMT models.

2.4 Data Augmentation

In this section, we provide the background of representative data augmentation techniques that are closely combined with data re-weighting to exploit the inter-sample quality information.

2.4.1 Back-Translation

Sequence-level knowledge distillation is the most popular data augmentation method in NLP. Taking the machine translation task as an example, back-translation (BT) [70] has been the most popular technique that has been widely applied to build large-scale NMT systems. Back-translation pairs each monolingual sentence at the target side with a synthetic sentence at the source side. Given a translation task from language $\mathcal{X}$ to $\mathcal{Y}$ with a bitext (X, Y), back-translation is performed in three steps: (1) Train a teacher translation model in the reversed direction, denoted as $g_T : \mathcal{Y} \rightarrow \mathcal{X}$. (2) Translate a set of monolingual data in the target language, denoted as Y_m, by the teacher model g_T, obtaining the synthetic data $(g_T(Y_m), Y_m)$. (3) Train a student translation model on the combination of the bitext

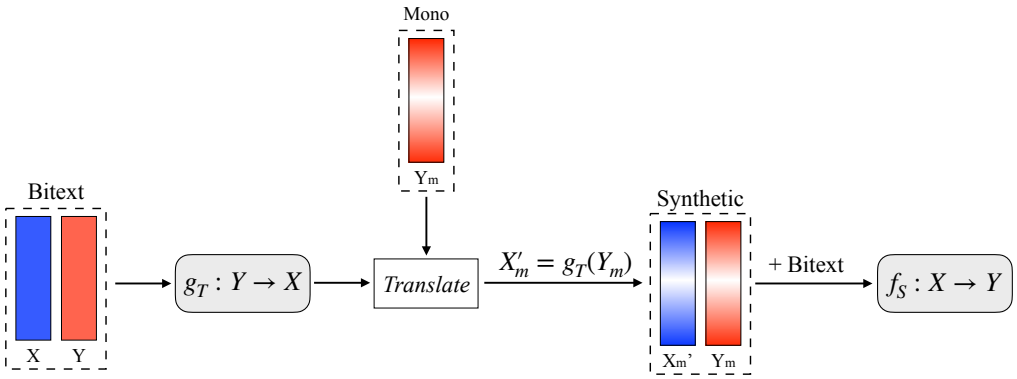
and the synthetic data as the final model. Figure 2.15(a) illustrates the process of back-translation.

As an effective approach, back-translation has been widely adopted and constantly improved. For example, to improve the efficiency of back-translation, Fadaee et al. [31] demonstrate that words with high prediction loss during training benefit most from the back-translation data, and thus propose to sample examples with difficult-to-predict words for back-translation. Wang et al. [32] point out that model predictions during back-translation are inevitably erroneous, which could impair the performance of the NMT models, and propose to quantify the confidence of the NMT model predictions based on model uncertainty. Edunov et al. [71] demonstrate that generating source sentences by sampling or beam search with noise introduced (i.e., **noised-BT**) can significantly improve the translation performance. Further, Edunov et al. [72] also find that the NMT systems trained with back-translation generate translations with satisfactory adequacy and high fluency, which are more favored by humans. Caswell et al. [73] propose **tagged-BT**, which prepends a unique tag to each synthetic source sentence of back-translation, and show that it plays a similar role as noise. More recently, Marie et al. [74] revisit tagged-BT and discover that the tag prevents back-translation from degrading the translation performance on test samples with natural text as source sentences.

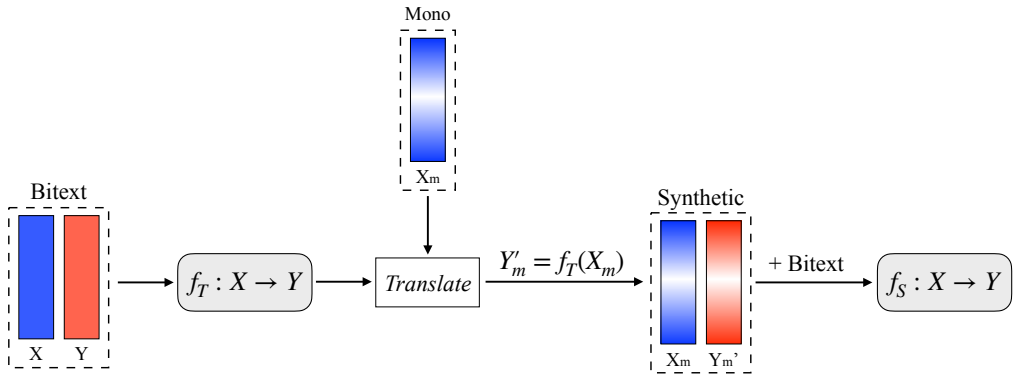
2.4.2 Self-Training

Self-training [75] shares a similar procedure as back-translation, but with two differences: (1) The teacher model is in the forward direction, i.e., $f_T : \mathcal{X} \rightarrow \mathcal{Y}$. (2) The monolingual data comes from the source language X. The workflow of self-training is illustrated in Figure 2.15(b).

While self-training has been extensively investigated on classification problems, in machine translation it is still unclear how self-training works due to the compositionality of the target space.



(a) Back-translation



(b) Self-training

Figure 2.15: Workflows of back-translation and self-training.

Recently, there are some efforts paid to this problem. For example, He et al. [38] find that the perturbation on the hidden states (i.e., dropout) is critical for self-training to benefit from the synthetic data, and propose to inject noise into the source of synthetic data (i.e., **noised-ST**). Wu et al. [76] combine noised self-training with noised back-translation to exploit the monolingual data at scale. Both studies follow a noised data training and a subsequent fine-tuning paradigm, which performs on par with or better than joint training. While these works suggest that synthetic data manipulation [71, 73] and training strategy optimization [76, 32, 38] can boost the self-training performance significantly, how to efficiently and effectively sample a subset from the large-scale monolingual data has not been well studied.

Algorithm 2: Data Diversification [77].

Input: A dataset $\mathcal{D} = (S, T)$, a diversification factor k, the number of rounds N
Output: A trained source-target translation model $\hat{M}_{S \to T}$
procedure TRAIN($\mathcal{D} = (S, T)$)
Train randomly initialized M on $\mathcal{D} = (S, T)$ until convergence
return M
procedure DATADIVERSE($\mathcal{D} = (S, T)$,k,N)
$\mathcal{D}_0 \leftarrow \mathcal{D}$
for $r \in 1, \ldots, N$ **do**
    $\mathcal{D}_r = (S_r, T_r) \leftarrow \mathcal{D}_{r-1}$
    for $i \in 1, \ldots, k$ **do**
        $M^i_{S \to T,r} \leftarrow \text{TRAIN}(\mathcal{D}_{r-1} = (S_{r-1}, T_{r-1}))$
        $M^i_{T \to S,r} \leftarrow \text{TRAIN}(\mathcal{D}'_{r-1} = (T_{r-1}, S_{r-1}))$
        $\mathcal{D}_r \leftarrow \mathcal{D}_r \bigcup (S, M^i_{S \to T,r}(S))$
        $\mathcal{D}_r \leftarrow \mathcal{D}_r \bigcup (M^i_{T \to S,r}(T), T)$
    end
end
$\hat{M}_{S \to T} \leftarrow \text{TRAIN}(\mathcal{D}_N)$
return $\hat{M}_{S \to T}$

2.4.3 Hybrid Approach

Recent studies also try to combine the benefits of both back-translation and self-training, for example, **data diversification**. Data diversification [77] aims at taking full advantage of the training data for the NMT models. It diversifies the training data by using the predictions of multiple forward and backward models and then merging them with the original dataset on which the final NMT model is trained. Essentially, data diversification utilizes the aforementioned back-translation and self-training techniques on the original training data to synthesize more diverse data. Formally, let $\mathcal{D} = (S, T)$ be the parallel training data, where S denotes the source-side corpus and T the target-side corpus. Also, let $\mathcal{M}_{S \to T}$ and $\mathcal{M}_{T \to S}$ represent the forward and backward NMT models, respectively. The forward NMT model $\mathcal{M}_{S \to T}$ translates the source-side corpus to obtain synthetic data $(S, \mathcal{M}_{S \to T}(S))$ and the backward NMT model translates the target-side corpus to obtain another synthetic data $(\mathcal{M}_{T \to S}(T), T)$.

For either forward or backward model, there are k different models where k denotes the diversification factor. For example, we can use k different random seeds to train the NMT models. Then, we can train new NMT models from scratch on the combination of the $2k$ copies of synthetic data and the original data. The process can be iterated with the improved forward and backward NMT models to generate better synthetic data. Algorithm 2 shows the details.

Regarding our studies on data augmentation, we rely on the self-training technique to rejuvenate inactive sample in our DataReju framework [41] to fully utilize the bilingual training data. In addition, we propose the UncSamp strategy [42] to further improve the performance of self-training in NMT.

2.5 Inference and Evaluation

In this section, we introduce the datasets, the inference methods, and the evaluation metrics involved in this thesis.

2.5.1 Datasets

Emotion Recognition in Conversations. We list the commonly used datasets for emotion recognition in conversations in Table 2.1. Below shows the details of each dataset.

- *Friends*[2]: The Friends emotion dataset [78] is created based on the Friends TV Scripts, with multiple speakers involved in the conversations. The dataset is built to contain 1,000 conversations split into 720, 80, 200 ones as the training set, validation set, and testing set, respectively. Each utterance in a conversation indicates one of the eight emotions, i.e., anger, joy, sadness, neutrality, surprise, disgust, fear, and non-neutral.
- *EmotionPush*[3]: The EmotionPush emotion dataset [78] is also created based on private conversations between friends on the

[2] http://doraemon.iis.sinica.edu.tw/emotionlines
[3] http://doraemon.iis.sinica.edu.tw/emotionlines

Table 2.1: Datasets for emotion recognition in conversations.

Dataset	Conversation			Utterances		
	Train	Valid	Test	Train	Valid	Test
IEMOCAP	91	24	31	3,569	721	1,208
Friends	720	80	200	10,561	1,178	2,764
EmotionPush	720	80	200	10,733	1,202	2,807
MELD	1,039	114	280	9,989	1,109	2,610
EmoryNLP	713	99	85	9,934	1,344	1,328
MOSEI	2,250	300	678	26,603	3,281	7,629
OpenSubtitle2016	58,360	3,186	3,297	2.4M	130K	134K

Dataset	Emotions				
	Anger	Joy	Sadness	Neutral	Others
IEMOCAP	1,090	1,627	1,077	1,704	0
Friends	759	1,710	498	6,530	5,006
EmotionPush	140	2,100	514	9,855	2,133
MELD	1,607	2,308	1,002	6,436	2,355
EmoryNLP	1,332	2,755	844	3,776	3,899
MOSEI	2,086	9,765	2,965	5,913	16,784

Facebook messenger collected by an App called EmotionPush. It is annotated in the same fashion as Friends, and also contains 720, 80, 200 conversations in the training set, validation set, and testing set, respectively.

- *IEMOCAP*[4]: The IEMOCAP dataset [79] contains the acts of 10 speakers in a dyadic conversation fashion, providing text, audio, and video features. We follow the previous work to use the first four sessions of transcripts as the training set, and the last one as the testing set. The validation set is extracted from the randomly-shuffled training set with a ratio of 80:20. Also, we focus on recognizing four emotion classes, i.e., anger, happiness, sadness, and neutrality.

- *EmoryNLP*[5]: The EmoryNLP dataset [80] is also built from the Friends TV Scripts, but the utterances are annotated by

[4] https://sail.usc.edu/iemocap/
[5] https://github.com/emorynlp/emotion-detection/

a different emotion set, including sad, mad, scared, powerful, peaceful, joyful, and neutral. It contains 897 scenes with 12,606 utterances, and is split into the training set, validation set, and testing set by scenes with the ratio of 713:99:85.

- *MELD*[6]: The MELD dataset is an enhanced and extended version of the Friends dataset by Poria et al. [81]. The utterances are re-annotated with the aid of video clips such that some neutral or non-neutral utterances are categorized into specific emotion classes.
- *MOSEI*[7]: The MOSEI dataset [82] is a multimodal dataset for sentiment analysis and emotion recognition, providing text, audio, and video features. We use the raw transcripts, where each utterance is labeled with one or several of the six emotions, i.e., happiness, sadness, anger, disgust, surprise, and fear. We decide the emotion of an utterance by the majority vote or the higher emotion intensity sum if there are more than one majority vote. The training set, validation set, and testing set consist of 2,250, 300, and 678 utterances each.
- *OpenSubtitle2016*[8]: We collect unlabeled conversation data from the open-source database named OpenSubtitle[9] [83], which contains a large amount of subtitles of movies and TV shows. Specifically, we retrieve the English subtitles throughout the year of 2016, including 25466 `.html` files. After preprocessing, we obtain around 60K conversations with over 2.0M utterances.

Machine Translation. Commonly used datasets for machine translation are mainly from WMT, and IWSLT, as listed in Table 2.2. Below shows the details of the translation tasks.

- *WMT*[10]: The Conference on Machine Translation (WMT) built on a series of annual workshops and conferences on machine

[6] https://github.com/SenticNet/MELD/
[7] http://immortal.multicomp.cs.cmu.edu/raw_datasets/
[8] http://opus.nlpl.eu/OpenSubtitles-v2018.php
[9] http://opus.nlpl.eu/OpenSubtitles-v2018.php
[10] http://www.statmt.org/wmt20/

Table 2.2: Datasets for machine translation. "⇒" denotes the translation task in the forward direction and "⇐" in the backward direction.

Dataset	Train	Valid		Test	
		⇒	⇐	⇒	⇐
WMT14 En-De	4.5M	3000		3003	
WMT14 En-Fr	35.5M	6003		3003	
WMT16 En-Ro	608K	1999		1999	
WMT19 En-De	36.8M	2998	2998	1997	2000
WMT20 En-Zh	22.1M	1997	2000	1418	2000
WMT20 NewsCrawl En	200M	–		–	
IWSLT14 En-De	160K	7283		6750	

translation, going back to 2006. Every year, the conference features a variety of shared tasks, including news translation, biomedical translation, similar language translation, evaluation metrics, quality estimation, chat translation, and so on. The conference also encourages individuals who participate in the shared tasks to submit research papers to share their approach with the community. We focus on the news translation tasks, such as WMT14 English-German (En-De), WMT14 English-French (En-Fr), WMT16 English-Romanian (En-Ro), WMT19 English-German (En-De), and WMT20 English-Chinese (En-Zh) tasks in this thesis. We also include the English monolingual data from NewsCrawl released by WMT20.

- *IWSLT*[11]: The International Conference on Spoken Language Translation (IWSLT) is an annual scientific conference, associated with an open evaluation campaign on spoken language translation, where both scientific papers and system descriptions are presented. The conference attempts to address challenges in simultaneous translation, consecutive translation and subtitling tasks, for real-time, low latency as well offline archival purposes and under low-resource, multilingual, and multimodal constraints. IWSLT14 English-German (En-De) task is commonly used in research papers to evaluate the machine translation systems for

[11] https://iwslt.org/2021/

Algorithm 3: The beam search algorithm [3].

$t \leftarrow 1$
$\mathcal{A} = \{\langle <\text{bos}>, 0\rangle\}$ ▷ The set of alive candidates
$\mathcal{F} = \{\}$ ▷ The set of finished candidates
while $t <$ max_length **do**
  $\mathcal{C} = \{\}$
  for $\langle y_0 \dots y_{t-1}, v\rangle \in \mathcal{A}$ **do**
    $\mathbf{p} \leftarrow \text{NMT}(y_0 \dots y_{t-1}, \mathbf{x})$
    for $w \in \mathcal{V}$ **do**
      $y_t \leftarrow w$
      $l \leftarrow \log(\mathbf{p}[w])$
      $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{\langle y_0 \dots y_t, v + l\rangle\}$
    end
  end
  $\mathcal{C} \leftarrow \text{TopK}(\mathcal{C}, k)$
  for $\langle y_0 \dots y_t, v\rangle \in \mathcal{C}$ **do**
    if $y_t == <\text{eos}>$ **then**
      $\mathcal{F} \leftarrow \mathcal{F} \bigcup \{\langle y_0 \dots y_t, v\rangle\}$
    else
      $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{\langle y_0 \dots y_t, v\rangle\}$
    end
  end
  $\mathcal{A} \leftarrow \text{TopK}(\mathcal{A}, k)$
  $\mathcal{F} \leftarrow \text{TopK}(\mathcal{F}, k)$
  $t \leftarrow t + 1$
end
$\langle y_0 \dots y_t, v\rangle \leftarrow \text{Top}(\mathcal{F})$
return $y_1 \dots y_t$

low-resource translation.

For all the datasets, we follow the official split of training, validation and testing sets for experiments. While such a fixed split may result in cherry-pick methods, it enables a fair comparison with previous studies. Besides, it is usually inconvenient to conduct cross-validation on language generation tasks, because the validation and testing sets are usually obtain by human annotations.

2.5.2 Inference

Maximum Probability Prediction. Given a classification model and an input sentence $\mathbf{x}$, we obtain the prediction by selecting the label with the maximum probability:

$$\tilde{\mathbf{y}} = \arg\max_{c \in \mathcal{C}} P(c|x = \mathbf{x}; \Theta), \tag{2.30}$$

where $\mathcal{C}$ denotes the possible classes of the label.

Beam Search. Given an NMT model and a source sentence $\mathbf{x}$, how to generate a translation from the model is an important problem. Ideally, we would like to find the target sentence $\mathbf{y}$ which maximizes the model prediction $P(\mathbf{y}|x = \mathbf{x}; \Theta)$ as the translation. However, due to the intractably large search space, it is impractical to find the translation with the highest probability. Therefore, NMT typically uses local search algorithms such as greedy search or beam search to find a local best translation. Beam search is a classic local search algorithm which has been widely used in NMT [3]. Previously, beam search has been successfully applied in statistical machine translation (SMT). The beam search algorithm keeps track of k states during the inference stage. Each state is a tuple $(y_0 \dots y_t, v)$, where $(y_0 \dots y_t)$ is a candidate translation, and v is the log-probability of the candidate. At each step, all the successors of all k states are generated, but only the top-k successors are selected. Each state will result in k partial translations after one-step generation, which is the so-called successors of the state. The algorithm usually terminates when the step exceeds a pre-defined value or k full translations are found. It should be noted that the beam search will degrade into the greedy search if $k = 1$. The pseudo-codes of the beam search algorithm are given in Algorithm 3.

2.5.3 Evaluation Metrics

We introduce the commonly used evaluation metrics for text classification and generation, i.e., F1-score and BLEU score, respectively.

F1-score. The performance of a classification model on a test dataset

		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	True Positive	False Negative
	Class=No	False Positive	True Negative

Figure 2.16: A confusion matrix of binary classification.

can be described by a confusion matrix. Take the binary classification task as an example, Fig. 2.16 shows the confusion matrix, in which True Positive (TP) and True Negative (TN) represent the observations that are correctly predicted. The goal of classification is to minimize False Positive (FP) and False Negative (FN).

Based on the confusion matrix, three metrics can be defined:

- *Precision* - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.31}$$

- *Recall* - Recall is the ratio of correctly predicted positive observations to the total observations in actual Class=Yes:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.32}$$

- *F1-score* - F1-score is the harmonic average of Precision and Recall. Therefore, F1-score takes both FP and FN into account, which is usually more useful than accuracy for imbalanced datasets. The formulation of F1-score is:

$$\text{F1} - \text{score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{2.33}$$

For multi-class classification tasks, we need to calculate precision, recall, and F1-score for each class in the test set. To measure the overall performance on all the classes, we use the following four metrics:

- *Weighted Accuracy*: The average Recall weighted by the percentage of true instances in each class.
- *Unweighted Accuracy*: The unweighted mean of Recall.
- *Weighted F1*: The average F1-score weighted by the percentage of true instances in each class.
- *Macro-averaged F1*: The unweighted mean of F1-score.

Generally, recognizing strong emotions like *happy* or *angry* is more meaningful than recognizing *neutral*. Therefore, the unweighted metrics are more favorable because the weighted ones are usually compromised by the majority emotion type *neutral*.

Top-K Recall. For text retrieval tasks, we usually evaluate them with the recall of the true positives among k best-matched answers from N' available candidates based on the representation of an utterance [84]. The metric is defined as below:

$$\mathbf{R}_{N'}@k = \frac{\sum_{i=1}^{k} y_i}{\sum_{i=1}^{N'} y_i}, \tag{2.34}$$

where the variate y_i represents the binary label for each candidate answer, i.e., 1 for the correct answer and 0 for the wrong answers. The metric is calculated for each sample and the average result of all the samples will be reported.

BLEU Score. Manual evaluation of machine translation outputs is not only expensive but also impractical to scaling for more languages. On the contrary, automatic evaluation is inexpensive and language-independent, with BLEU [85] as the representative automatic evaluation metric.

BLEU stands for bilingual evaluation understudy, which is an algorithm to evaluate the quality of machine translations. Specifically, BLEU is computed using a couple of n-gram modified precisions to compare a candidate translation against multiple reference transla-

tions. The expression of BLEU is as below [85]:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{2.35}$$

where p_n is the modified precision for n-gram, w_n is the weight for summarizing different n-gram precision $\log p_n$ and $\sum_{n=1}^{N} w_n = 1$, and BP denotes the brevity penalty to penalize short machine translations. BP is defined as below [85]:

$$\text{BP} = \begin{cases} 1, & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right), & \text{if } c \leq r \end{cases}, \tag{2.36}$$

where c is the number of unigrams in all the candidate sentences, r is the best match lengths for each candidate sentence in the corpus. Here the best match length is the closest reference sentence length to the candidate sentences. For example, if there are three references with lengths 12, 14, and 17 words and the candidate translation is a terse 13 words, ideally the best match length could be either 12 or 14, but we arbitrarily choose the shorter one which is 12.

Usually, BLEU is evaluated on the corpus where there are many candidate sentences translated from different source texts and each of them has several reference sentences. Then c is the total number of unigrams in all the candidate sentences, and r is the sum of the best match lengths for each candidate sentence in the corpus.

□ **End of chapter.**

Chapter 3

Intra-Sample Structure Mining for Context Enhancement

In this chapter, we investigate the intra-sample structure for context enhancement. We conduct the studies on the ERC task due to the rich structure information of conversations. For the ERC task, context is important as the same sentence can deliver different emotions in different contexts. Besides, existing approaches are weak at capturing long-range context. Therefore, we propose a Hierarchical Gated Recurrent Unit (HiGRU) framework to learn both the context of words and that of utterances. Moreover, we promote the framework to two variants, HiGRU with individual features fusion (HiGRU-f) and HiGRU with self-attention and features fusion (HiGRU-sf), so that the word/utterance-level individual inputs and the long-range contextual information can be sufficiently utilized. Experiments on three conversation emotion datasets, IEMOCAP, Friends, and EmotionPush demonstrate that our proposed HiGRU models attain significant improvements over the state-of-the-art methods.

3.1 Problems and Motivation

Instead of investigating sentence-level language understanding tasks, we choose the ERC task because of the rich context in conversations, which is important for understanding accurately. Besides, emotion recognition itself is also a significant artificial intelligence research

Table 3.1: The word "okay" exhibits different emotions in the American television sitcom, Friends.

Speaker	Utterance	Emotion
Rachel	Oh okay, I'll fix that to. What's her email address?	Neutral
Ross	Rachel!	Anger
Rachel	All right, I promise. I'll fix this. I swear. I'll-I'll- I'll-I'll talk to her.	Non-neutral
Ross	**Okay!**	**Anger**
Rachel	**Okay.**	**Neutral**
Nurse	This room's available.	Neutral
Rachel	**Okay!**	**Joy**
Rachel	Okay wait!	Non-neutral
Rachel	You listen to me!	Anger

topic due to the promising potential of developing empathetic machines for people.

Emotion is a universal phenomenon across different cultures and mainly consists of six basic types: anger, disgust, fear, happiness, sadness, and surprise [86, 87]. While the ERC task can also be performed for each utterance in the conversations individually, capturing the context between utterances provides more information to make the right predictions. Otherwise, the same word or sentence can deliver different emotions in different contexts. For example, in Table 3.1, the sentence "okay" can deliver three different emotions, anger, neutral, and joy, respectively. Strong emotions like joy and anger may be indicated by the symbols "!" or "?" along with the word. To identify a speaker's emotion precisely, we need to explore the conversation context sufficiently. At the meantime, existing context-dependent models like cLSTM [5] still have room for further improvements: 1) The textual feature is extracted by CNN, which should be incompatible with the upper-level RNNs and could be replaced by a bidirectional RNN. 2) The upper-level RNNs alone are weak in capturing long-range context and weighting the importance

of each context.

Incorporating all these factors, in the chapter, we propose a Hierarchical Gated Recurrent Unit (HiGRU) framework for the utterance-level emotion recognition in conversation. More specifically, HiGRU is composed of two levels of bidirectional GRUs: a lower-level GRU to model the word sequences of each utterance to produce individual utterance embeddings, and an upper-level GRU to capture the sequential and contextual relationship of utterances. We further promote the proposed HiGRU to two variants: HiGRU with individual features fusion (HiGRU-f), and HiGRU with self-attention and features fusion (HiGRU-sf). In HiGRU-f, the individual inputs, i.e., the word embeddings in the lower-level GRU and the individual utterance embeddings in the upper-level GRU, are concatenated with the hidden states to generate the contextual word/utterance embeddings, respectively. In HiGRU-sf, a self-attention layer is placed on the hidden states from the GRU to learn long-range contextual embeddings, which are concatenated with the original individual embeddings and the hidden states to generate the contextual word/utterance embeddings. Finally, the contextual utterance embedding is sent to a fully connected layer to determine the corresponding emotion. To alleviate the effect of the data imbalance issue, we also follow [88] to train our models by minimizing a weighted categorical cross-entropy. We summarize our contributions as follows:

- We propose a HiGRU framework to better learn both the individual utterance embeddings and the contextual information of utterances, so as to recognize the emotions more precisely.
- We propose two progressive HiGRU variants, HiGRU-f and HiGRU-sf, to effectively incorporate the individual word- and utterance-level information and the long-range contextual information respectively.
- We conduct extensive experiments on three textual conversation emotion datasets, IEMOCAP, Friends, and EmotionPush. The results demonstrate that our proposed HiGRU models achieve

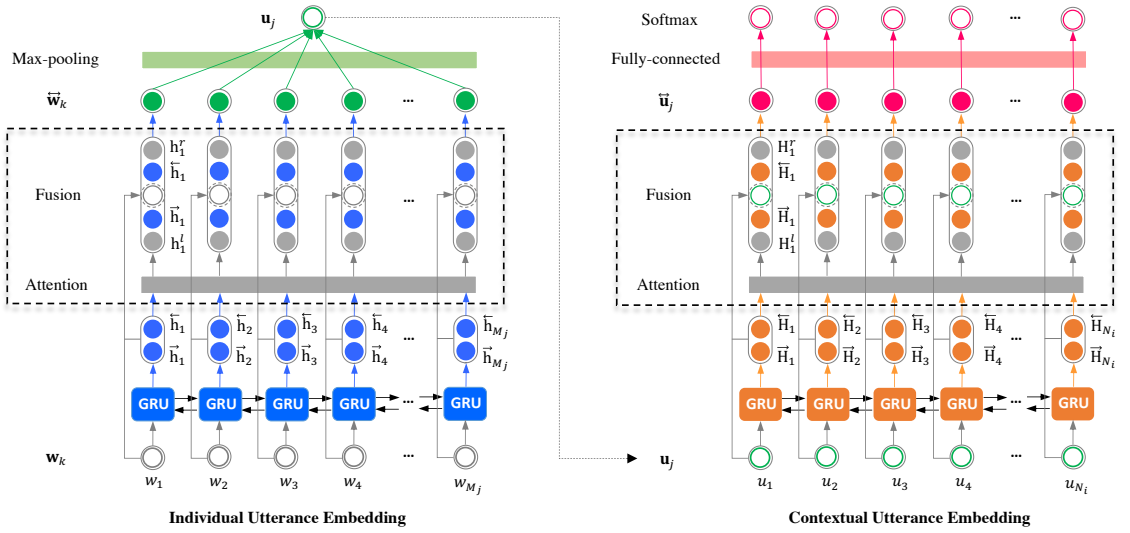


Figure 3.1: The architecture of our proposed HiGRU-sf. "Attention" denotes self-attention. By removing the "Attention" layer, we attain HiGRU-f, and by further removing the "Fusion" layer, we can recover the vanilla HiGRU.

at least 8.7%, 7.5%, 6.0% improvement over state-of-the-art methods on each dataset, respectively. Particularly, by utilizing only the textual feature in IEMOCAP, our proposed HiGRU models gain at least 3.8% improvement over the existing best model, conversational memory network (CMN) with not only the text feature, but also the visual, and audio features.

3.2 Methodology

3.2.1 Vanilla HiGRU

The vanilla HiGRU consists of two-level GRUs: the lower-level bidirectional GRU is to learn the individual utterance embedding by modeling the word sequence within an utterance and the upper-level bidirectional GRU is to learn the contextual utterance embedding by modeling the utterance sequence within a conversation.

Individual Utterance Embedding. For the j^{th} utterance in the i^{th} conversation C_i, it is represented by a sequence of word vectors, $\mathbf{x}_j = \{\mathbf{w}_k\}_{k=1}^{M_j}$, where M_j is the number of words. The word vectors are fed into the lower-level bidirectional GRU [47] to learn the individual

utterance embedding in two opposite directions:

$$\overrightarrow{\mathbf{h}}_k = \text{GRU}(\mathbf{w}_k, \overrightarrow{\mathbf{h}}_{k-1}), \tag{3.1}$$

$$\overleftarrow{\mathbf{h}}_k = \text{GRU}(\mathbf{w}_k, \overleftarrow{\mathbf{h}}_{k+1}). \tag{3.2}$$

The two hidden states $\overrightarrow{\mathbf{h}}_k$ and $\overleftarrow{\mathbf{h}}_k$ are concatenated into $\overleftrightarrow{\mathbf{h}}_k = [\overrightarrow{\mathbf{h}}_k; \overleftarrow{\mathbf{h}}_k]$ to produce the *contextual word embedding* via the $\tanh(\cdot)$ activation function on a linear transformation:

$$\overleftrightarrow{\mathbf{w}}_k = \tanh(\mathbf{W}_u \cdot \overleftrightarrow{\mathbf{h}}_k + \mathbf{b}_u), \tag{3.3}$$

where $\mathbf{W}_u \in \mathbb{R}^{d_1 \times 2d_1}$ and $\mathbf{b}_u \in \mathbb{R}^{d_1}$ are the model parameters, d_0 and d_1 are the dimensions of word embeddings and the hidden states of the lower-level GRU, respectively.

To extract the features of the utterance, we follow previous studies [1] to apply max-pooling on the contextual word embeddings within the utterance. The obtained individual utterance embeddings is expressed as below:

$$\mathbf{u}_j = \text{maxpool}\left(\{\overleftrightarrow{\mathbf{w}}_k\}_{k=1}^{M_j}\right). \tag{3.4}$$

Contextual Utterance Embedding. For the i^{th} conversation, $C_i = \{(u_j, s_j, c_j)\}_{j=1}^{N_i}$, the learned individual utterance embeddings, i.e., $\{\mathbf{u}_j\}_{j=1}^{N_i}$, are fed into the upper-level bidirectional GRU to capture the sequential and contextual relationship of utterances in a conversation:

$$\overrightarrow{\mathbf{H}}_j = \text{GRU}(\mathbf{u}_j, \overrightarrow{\mathbf{H}}_{j-1}), \tag{3.5}$$

$$\overleftarrow{\mathbf{H}}_j = \text{GRU}(\mathbf{u}_j, \overleftarrow{\mathbf{H}}_{j+1}). \tag{3.6}$$

Here, the hidden states of the upper-level GRU are represented by $\mathbf{H}_j \in \mathbb{R}^{d_2}$, where d_2 is the dimension of the hidden states of the upper-level GRU, to distinguish from those learned in the lower-level GRU denoted by $\mathbf{h}_k$. Accordingly, we can obtain the *contextual utterance embedding* by:

$$\overleftrightarrow{\mathbf{u}}_j = \tanh(\mathbf{W}_c \cdot \overleftrightarrow{\mathbf{H}}_j + \mathbf{b}_c), \tag{3.7}$$

where $\overleftrightarrow{\mathbf{H}}_j = [\overrightarrow{\mathbf{H}}_j; \overleftarrow{\mathbf{H}}_j]$, $\mathbf{W}_c \in \mathbb{R}^{d_2 \times 2d_2}$ and $\mathbf{b}_c \in \mathbb{R}^{d_2}$ are the model parameters, and d_2 is the dimension of the hidden states in the upper-level GRU. Since the emotions are recognized at utterance-level, the learned contextual utterance embedding $\overleftrightarrow{\mathbf{u}}_j$ is directly fed to a FC layer followed by a softmax function to determine the corresponding emotion label:

$$\hat{\mathbf{y}}_j = \text{softmax}(\mathbf{W}_{fc} \cdot \overleftrightarrow{\mathbf{u}}_j + \mathbf{b}_{fc}), \tag{3.8}$$

where $\hat{\mathbf{y}}_j$ is the predicted vector over all emotions, and $\mathbf{W}_{fc} \in \mathbb{R}^{|\mathcal{C}| \times d_2}$, $\mathbf{b}_{fc} \in \mathbb{R}^{|\mathcal{C}|}$.

3.2.2 Individual Features Fusion

The vanilla HiGRU contains two main issues: 1) the individual word and utterance embeddings are diluted with the stacking of layers, i.e., the loss of low-level information; 2) the upper-level GRU tends to gather more contextual information from the majority emotions, which deteriorates the overall model performance.

To resolve these two problems, we propose to fuse individual word and utterance embeddings with the hidden states from GRUs so as to strengthen the information of each word and utterance in their corresponding contextual embeddings. Such a fusion operation is essentially the residual connection utilized in Transformer models [15], which proves to be effective to prevent gradient vanishing during training. We name this variant as **HiGRU-f**, representing HiGRU with individual features fusion. Hence, the lower-level GRU can maintain individual word embeddings and the upper-level GRU can relieve the effect of majority emotions and attain a more precise utterance representation for different emotions. Specifically, the contextual embeddings are updated as:

$$\overleftrightarrow{\mathbf{w}}_k = \tanh(\mathbf{W}_u \cdot \overleftrightarrow{\mathbf{h}}^f_k + \mathbf{b}_u), \tag{3.9}$$

$$\overleftrightarrow{\mathbf{u}}_j = \tanh(\mathbf{W}_c \cdot \overleftrightarrow{\mathbf{H}}^f_j + \mathbf{b}_c), \tag{3.10}$$

where $\mathbf{W}_u \in \mathbb{R}^{d_1 \times (d_0 + 2d_1)}$, $\mathbf{W}_c \in \mathbb{R}^{d_2 \times (d_1 + 2d_2)}$, $\overleftrightarrow{\mathbf{h}}^f_k = [\overrightarrow{\mathbf{h}}_k; \mathbf{w}_k; \overleftarrow{\mathbf{h}}_k]$, and $\overleftrightarrow{\mathbf{H}}^f_j = [\overrightarrow{\mathbf{H}}_j; \mathbf{u}_j; \overleftarrow{\mathbf{H}}_j]$.

3.2.3 Long-Range Context

Another challenging issue is to extract the contextual information of long sequences, especially the sequences in the testing set that are longer than those in the training set [89]. To fully utilize the global contextual information, we place a self-attention layer upon the hidden states of HiGRU and fuse the attention outputs with the individual word and utterance embeddings and the hidden states to learn the contextual word and utterance embeddings. Hence, this variant is termed **HiGRU-sf**, representing HiGRU with self-attention and features fusion.

Particularly, we apply self-attention upon the forward and backward hidden states separately to produce the left context embedding, $\mathbf{h}_k^l$ ($\mathbf{H}_j^l$), and the right context embedding, $\mathbf{h}_k^r$ ($\mathbf{H}_j^r$), respectively. This allows us to gather the unique global contextual information at the current step in two opposite directions and yield the corresponding contextual embeddings computed as follows:

$$\overleftrightarrow{\mathbf{w}}_k = \tanh(\mathbf{W}_u \cdot \overleftrightarrow{\mathbf{h}}_k^{sf} + \mathbf{b}_u), \tag{3.11}$$

$$\overleftrightarrow{\mathbf{u}}_j = \tanh(\mathbf{W}_c \cdot \overleftrightarrow{\mathbf{H}}_j^{sf} + \mathbf{b}_c), \tag{3.12}$$

where $\mathbf{W}_u \in \mathbb{R}^{d_1 \times (d_0 + 4d_1)}$ and $\mathbf{W}_c \in \mathbb{R}^{d_2 \times (d_1 + 4d_2)}$ are trainable weights, and $\overleftrightarrow{\mathbf{h}}_k^{sf}$ denotes $[\mathbf{h}_k^l; \overrightarrow{\mathbf{h}}_k; \mathbf{w}_k; \overleftarrow{\mathbf{h}}_k; \mathbf{h}_k^r]$, and $\overleftrightarrow{\mathbf{H}}_j^{sf}$ represents $[\mathbf{H}_j^l; \overrightarrow{\mathbf{H}}_j; \mathbf{u}_j; \overleftarrow{\mathbf{H}}_j; \mathbf{H}_j^r]$.

Self-Attention. Self-attention is an effective non-recurrent architecture to compute the relation between one input to all other inputs and has been successfully applied to various NLP applications such as reading comprehension [90], and neural machine translation [15]. Figure 3.2 shows the dot-product self-attention over the forward hidden states of GRU to learn the left context $\mathbf{h}_k^l$. Each element in the attention matrix is computed by:

$$f(\overrightarrow{\mathbf{h}}_k, \overrightarrow{\mathbf{h}}_p) = \begin{cases} \overrightarrow{\mathbf{h}}_k^\top \overrightarrow{\mathbf{h}}_p, & \text{if } k, p \leq M_j \\ -\infty, & \text{otherwise} \end{cases}, \tag{3.13}$$

An attention mask is then applied to waive the inner attention between the sequence inputs and paddings. At each step, the corresponding left

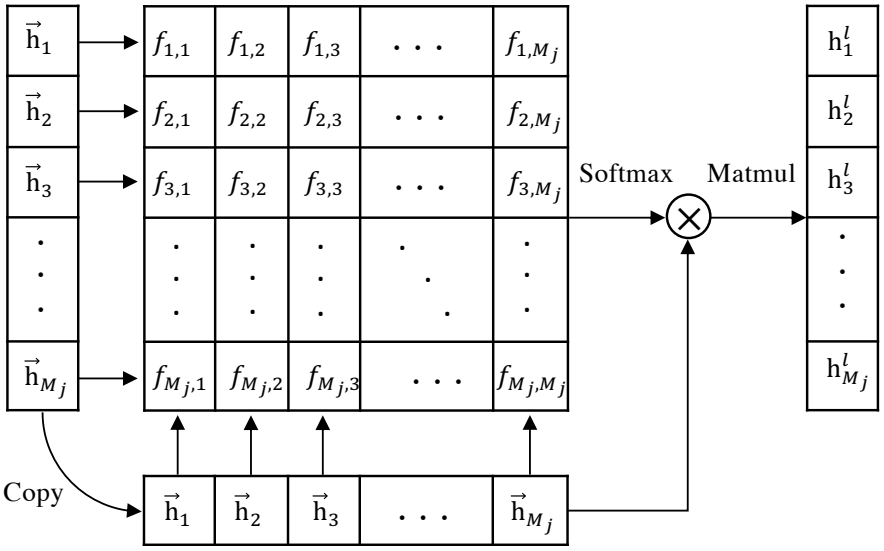


Figure 3.2: Self-attention over the forward hidden states of GRU.

context $\mathbf{h}_k^l$ is then computed by the weighted sum of all the forward hidden states:

$$\mathbf{h}_k^l = \sum_{p=1}^{M_j} a_{kp} \overrightarrow{\mathbf{h}}_p, \tag{3.14}$$

$$a_{kp} = \frac{\exp(f(\overrightarrow{\mathbf{h}}_k, \overrightarrow{\mathbf{h}}_p))}{\sum_{p'=1}^{M_j} \exp\Big(f(\overrightarrow{\mathbf{h}}_k, \overrightarrow{\mathbf{h}}_{p'})\Big)}, \tag{3.15}$$

where a_{kp} is the weight of $\overrightarrow{\mathbf{h}}_p$ to be included in $\mathbf{h}_k^l$. The right context $\mathbf{h}_k^r$ can be computed similarly.

3.2.4 Training Objective

Following [88] which attains the best performance in the EmotionX shared task [78], we minimize a weighted categorical cross-entropy on each utterance of all conversations to optimize the model parameters:

$$\mathcal{L} = -\frac{1}{\sum_{i=1}^{L} N_i} \sum_{i=1}^{L} \sum_{j=1}^{N_i} \omega(c_j) \sum_{c=1}^{|\mathcal{C}|} \mathbf{y}_j^c \log_2(\hat{\mathbf{y}}_j^c), \tag{3.16}$$

where $\mathbf{y}_j$ is the original one-hot vector of the emotion labels, and $\mathbf{y}_j^c$ and $\hat{\mathbf{y}}_j^c$ are the elements of $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$ corresponding to the class c.

Similar to [88], we assign the loss weight $\omega(c_j)$ inversely proportional to the number of training utterances in the class c_j, denoted by

Table 3.2: Statistics of the textual conversation datasets.

Dataset	Emotion				
	Ang	Hap/Joy	Sad	Neu	Others
IEMOCAP	1,090	1,627	1,077	1,704	0
Friends	759	1,710	498	6,530	5,006
EmotionPush	140	2,100	514	9,855	2,133

I_c, i.e., assigning larger loss weights for the minority classes to relieve the data imbalance issue. The difference is that we add a constant α to adjust the smoothness of the distribution. Then, we have:

$$\frac{1}{\omega(c)} = \frac{I_c^\alpha}{\sum_{c'=1}^{|\mathcal{E}|} I_{c'}^\alpha}. \tag{3.17}$$

3.3 Experiment

We conduct systematical experiments to demonstrate the advantages of our proposed HiGRU models.

3.3.1 Datasets

The experiments are carried out on three textual conversation emotion datasets (see the statistics in Table 3.2):

IEMOCAP. The IEMOCAP[1] dataset contains approximately 12 hours of audiovisual data, including video, speech, motion capture of face, text transcriptions. Following [5, 6]: (1) We apply the first four sessions for training and the last session for testing; (2) The validation set is extracted from the shuffled training set with the ratio of 80:20; (3) We only evaluate the performance on four emotions: anger, happiness, sadness, neutral, and remove the remaining utterances.

Friends. The Friends[2] dataset is annotated from the Friends TV Scripts [78], where each conversation in the dataset consists of a scene of multiple speakers. Totally, there are 1,000 conversations, which

[1] https://sail.usc.edu/iemocap/
[2] http://doraemon.iis.sinica.edu.tw/emotionlines

are split into 720, 80, and 200 conversations for training, validation, and testing, respectively. Each utterance in a conversation is labeled by one of the eight emotions: anger, joy, sadness, neutral, surprise, disgust, fear, and non-neutral.

EmotionPush. The EmotionPush[3] dataset consists of private conversations between friends on the Facebook messenger collected by an App called EmotionPush, which is released for the EmotionX shared task [78]. Totally, there are 1,000 conversations, which are split into 720, 80, 200 conversations for training, validation, and testing, respectively. All the utterances are categorized into one of the eight emotions as in the Friends dataset.

Following the setup of Hsu et al. [78], in Friends and EmotionPush, we only evaluate the model performance on four emotions: anger, joy, sadness, and neutral, and exclude the contribution of the remaining emotion classes during training by setting their loss weights to zero.

Data Preprocessing. We preprocess the datasets by the following steps: (1) The utterances are split into tokens with each word being made into the lowercase; (2) All non-alphanumerics except "?" and "!" are removed because these two symbols usually exhibit strong emotions, such as surprise, joy and anger; (3) We clip the length of each utterance to 60 tokens and apply padding to utterances shorter than 60 tokens. (4) We build a dictionary based on the words and symbols extracted, and follow [5] to represent the tokens by the publicly available 300-dimensional word2vec[4] vectors trained on 100 billion words from Google News. The tokens not included in the word2vec dictionary are initialized by randomly generated vectors.

[3] http://doraemon.iis.sinica.edu.tw/emotionlines
[4] https://code.google.com/archive/p/word2vec/

3.3.2 Evaluation Metrics

To conduct fair comparison, we adopt two metrics as [78], the weighted accuracy (WA) and unweighted accuracy (UWA):

$$\text{WA} = \sum_{c=1}^{|\mathcal{C}|} p_c \cdot a_c, \tag{3.18}$$

$$\text{UWA} = \frac{1}{|\mathcal{C}|} \sum_{c=1}^{|\mathcal{C}|} a_c, \tag{3.19}$$

where p_c is the percentage of the class c in the testing set, and a_c is the corresponding accuracy.

Generally, recognizing strong emotions may provide more value than detecting the neutral emotion [78]. Thus, in Friends and EmotionPush, UWA is a more favorite evaluation metric because WA is heavily compromised with a large proportion of the neutral emotion.

3.3.3 Compared Methods

Our proposed vanilla HiGRU, HiGRU-f, and HiGRU-sf[5] are compared with the following state-of-the-art baselines:

- bcLSTM [5]: a bidirectional contextual LSTM with multimodal features extracted by CNNs.
- CMN [6]: a conversational memory network with multimodal features extracted by CNNs.
- SA-BiLSTM [91]: a self-attentive bidirectional LSTM model, a neat model achieving the second place of EmotionX Challenge [78].
- CNN-DCNN [88]: a convolutional-deconvolutional autoencoder with more handmade features, the winner of EmotionX Challenge [78].
- bcLSTM$_*$ and bcGRU: our implemented bcLSTM and bcGRU with the weighted loss on the textual feature extracted from

[5] https://github.com/wxjiao/HiGRUs

CNNs. We use $_*$ for our bcLSTM to distinguish from the results of the original bcLSTM.

3.3.4 Training Procedure

All our implementations are coded on the Pytorch framework. To prevent the models from fitting the order of data, we randomly shuffle the training set at the beginning of every epoch.

Parameters. For bcLSTM$_*$ and bcGRU, the CNN layer follows the setup of Kim et al. [1], i.e., consisting of the kernels of 3, 4, and 5 with 100 feature maps each. The convolution results of each kernel are fed to a max-over-time pooling operation. The dimension of the hidden states of the upper-level bidirectional LSTM or GRU is set to 300. For HiGRU, HiGRU-f, and HiGRU-sf, the dimensions of hidden states are set to 300 for both levels. The final FC layer contains two sub-layers with 100 neurons each.

Training. We adopt Adam [92] as the optimizer and set an initial learning rate, 1×10^{-4} for IEMOCAP and 2.5×10^{-4} for Friends and EmotionPush, respectively. An annealing strategy is utilized by decaying the learning rate by half every 20 epochs. Early stopping with a patience of 10 is adopted to terminate training based on the accuracy of the validation set. Specifically, following the best models on each dataset, the parameters are tuned to optimize WA on the validation set of IEMOCAP and to optimize UWA on the validation set of Friends and EmotionPush, respectively. Gradient clipping with a norm of 5 is applied to model parameters. To prevent overfitting, dropout with a rate of 0.5 is applied after the contextual word/utterance embeddings, and the FC layer.

Loss weights. For Friends and EmotionPush, as mentioned in Section 3.3.1, the loss weights are set to zero except the four considered emotions, to ignore the others during training. Besides, the power rate α of loss weights is tested from 0 to 1.5 with a step of 0.25, and we use the best one for each model and dataset.

Table 3.3: Experimental results on IEMOCAP. "(Feat)" represents the features used in the models, where T, V, and A denote the textual, visual, and audio features, respectively. The underlined results of bcLSTM and CMN are derived by us accordingly, while "-" represents that the results are unavailable from the original paper. For each emotion class, the corresponding column of values are the accuracy scores.

Model (Feat)	**Ang**	**Hap**	**Sad**	**Neu**	**WA**	**UWA**
bcLSTM [5] (T)	76.07	78.97	76.23	67.44	73.6	74.6
(T+V+A)	77.98	79.31	78.30	69.92	76.1	76.3
CMN [6] (T)	-	-	-	-	74.1	-
(T+V+A)	**89.88**	81.75	77.73	67.32	77.6	79.1
bcLSTM$_*$ (T)	75.29	79.40	78.07	76.53	77.7$_{(1.1)}$	77.3$_{(1.4)}$
bcGRU (T)	77.20	80.99	76.26	72.50	76.9$_{(1.6)}$	76.7$_{(1.3)}$
HiGRU (T)	75.41	**91.64**	79.79	70.74	80.6$_{(0.5)}$	79.4$_{(0.5)}$
HiGRU-f (T)	76.69	88.91	80.25	75.92	81.5$_{(0.7)}$	80.4$_{(0.5)}$
HiGRU-sf (T)	74.78	89.65	**80.50**	**77.58**	**82.1**$_{(0.4)}$	**80.6**$_{(0.2)}$

3.3.5 Main Results

Table 3.3 and Table 3.4 report the average results of 10 trials each on the three datasets, where the standard deviations of WA and UWA are recorded by the subscripts in round brackets. The results of bcLSTM, CMN, SA-BiLSTM, and CNN-DCNN are copied directly from the original papers for a fair comparison because we follow the same configuration for the corresponding datasets. From the results, we have several findings elaborated as below.

Baselines. The baseline models implemented us, i.e., bcLSTM$_*$ and bcGRU, attain comparable performance with the state-of-the-art methods on all three datasets. From the results on IEMOCAP in Table 3.3, we can observe that:

- By utilizing the textual feature only, bcGRU outperforms bcLSTM and CMN trained on the textual feature significantly, attaining +3.3 and +2.8 gain in terms of WA, respectively. bcLSTM$_*$ performs better than bcGRU, and even beats bcLSTM and CMN with the trimodal features in terms of WA. In terms of UWA, CMN performs better than bcLSTM$_*$ only when it is equipped

Table 3.4: Experimental results on Friends and EmotionPush. In the Train column, F(E) denotes the model is trained on only one training set, Friends or EmotionPush. F+E means the model is trained on the mixed training set while validated and tested individually.

Model	**Train**	**Friends (F)**					
		Ang	Joy	Sad	Neu	WA	UWA
SA-BiLSTM [91]	F+E	49.1	68.8	30.6	**90.1**	-	59.6
CNN-DCNN [88]	F+E	55.3	71.1	55.3	68.3	-	62.5
bcLSTM$_*$	F	64.7	69.6	48.0	75.6	72.4(4.2)	64.4(1.6)
bcGRU	F	69.5	65.4	52.9	74.7	71.7(4.7)	65.6(1.2)
bcLSTM$_*$	F+E	54.5	75.6	43.4	73.0	70.5(4.5)	61.6(1.6)
bcGRU	F+E	59.0	78.6	42.3	71.4	70.2(5.1)	62.8(1.4)
HiGRU	F	66.9	73.0	51.8	77.2	**74.4**(1.7)	67.2(0.6)
HiGRU-f	F	69.1	72.1	**60.4**	72.1	71.3(2.9)	68.4(1.0)
HiGRU-sf	F	**70.7**	70.9	57.7	76.2	74.0(1.4)	**68.9**(1.5)
HiGRU	F+E	55.4	81.2	51.4	64.4	65.8(4.2)	63.1(1.5)
HiGRU-f	F+E	54.9	78.3	55.5	68.7	68.5(3.0)	64.3(1.2)
HiGRU-sf	F+E	56.8	**81.4**	52.2	68.7	69.0(2.0)	64.8(1.3)

Model	**Train**	**EmotionPush (E)**					
		Ang	Joy	Sad	Neu	WA	UWA
SA-BiLSTM [91]	F+E	24.3	70.5	31.0	**94.2**	-	55.0
CNN-DCNN [88]	F+E	45.9	76.0	51.7	76.3	-	62.5
bcLSTM$_*$	E	32.9	69.9	47.1	78.0	74.7(4.4)	57.0(2.1)
bcGRU	E	33.7	71.1	57.2	76.1	73.9(2.9)	59.5(1.8)
bcLSTM$_*$	F+E	52.4	79.1	54.7	73.3	73.4(3.8)	64.9(2.1)
bcGRU	F+E	49.4	74.8	61.9	72.4	72.1(4.3)	64.6(1.8)
HiGRU	E	55.6	78.1	57.4	73.8	73.8(2.0)	66.3(1.7)
HiGRU-f	E	55.9	78.9	60.4	72.4	73.0(2.2)	66.9(1.2)
HiGRU-sf	E	57.5	78.4	64.1	72.5	73.0(1.6)	68.1(1.2)
HiGRU	F+E	50.8	76.9	69.0	75.7	75.3(1.7)	68.1(1.2)
HiGRU-f	F+E	**58.3**	79.1	**69.6**	70.0	71.5(2.5)	69.2(0.9)
HiGRU-sf	F+E	57.8	**79.3**	66.3	77.4	**77.1**(1.0)	**70.2**(1.1)

with multimodal features.

- By examining the detailed accuracy in each emotion, bcLSTM$_*$ and bcGRU with the textual feature attain much higher accuracy on the neutral emotion than bcLSTM with the only textual

feature while maintaining good performance on the other three emotions. The results show that the weighted loss function benefits the training of models.

From the results on Friends and EmotionPush in Table 3.4, we observe that bcLSTM$_*$ and bcGRU trained on the same dataset (F+E) of CNN-DCNN perform better than CNN-DCNN on EmotionPush while attaining comparable performance with CNN-DCNN on Friends. The results show that by utilizing the contextual information with the weighted loss function, bcLSTM$_*$ and bcGRU can beat the state-of-the-art method.

HiGRUs vs. Baselines. Our proposed HiGRUs outperform the state-of-the-art methods with significant margins on all the datasets. From Table 3.3, we observe that:

- CMN with the trimodal features attains the best performance on the anger emotion while our vanilla HiGRU achieves the best performance on the happiness emotion and gains further improvement on sadness and neutral emotions over CMN. Overall, the vanilla HiGRU achieves at least 8.7% and 3.8% improvement over CMN with the textual feature and the trimodal features in terms of WA, respectively. The results, including those of bcLSTM$_*$ and bcGRU, indicate that GRU learns better representations of utterances than CNN in this task.
- The two variants, HiGRU-f and HiGRU-sf, can further attain +0.9 and +1.5 improvement over HiGRU in terms of WA and +1.0 and +1.2 improvement over HiGRU in terms of UWA, respectively. The results demonstrate that the included individual word/utterance-level features and long-range contextual information in HiGRU-f and HiGRU-sf, are indeed capable of boosting the performance of the vanilla HiGRU.

As for the results in Table 3.4, we can see that:

- In terms of UWA, HiGRU trained and tested on individual sets of Friends and EmotionPush gains at least 7.5% and 6.0% im-

provement over CNN-DCNN, respectively. Overall, our proposed HiGRU achieves well-balanced performance for the four tested emotions, especially attaining significantly better performance on the minority emotions of anger and sadness.

- Moreover, HiGRU-f and HiGRU-sf further improve HiGRU +1.2 accuracy and +1.7 accuracy on Friends and +0.6 accuracy and +1.8 accuracy on EmotionPush in terms of UWA, respectively. The results again demonstrate the superior power of HiGRU-f and HiGRU-sf.

Mixing Training Sets. By examining the results from the last ten rows in Table 3.4, we conclude that it does not necessarily improve the performance by mixing the two sets of training data. Though the best performance of SA-BiLSTM and CNN-DCNN is obtained by training on the mixed dataset, the testing results show that our implemented bcLSTM$_*$, bcGRU and our proposed HiGRU models can attain better performance on EmotionPush but yield worse performance on Friends in terms of UWA.

By examining the detailed emotions, we speculate that: EmotionPush is a highly imbalanced dataset with over 60% of utterances in the neutral emotion. Introducing EmotionPush into a more balanced dataset, Friends, is equivalent to down-sampling the minority emotions in Friends. This hurts the performance on the minority emotions, anger and sadness. Meanwhile, introducing Friends into EmotionPush corresponds to up-sampling the minority emotions in EmotionPush. The performance of the sadness emotion is significantly boosted and that on the anger emotion is at least unaffected.

3.4 Analysis

3.4.1 Model Size

We study how the scale of the utterance encoder affects the performance of our proposed models, especially when our models contain a similar number of parameters as the baseline, say bcGRU. Such a fair

Table 3.5: Experimental results of UWA on Friends by our proposed models with different scales of utterance encoder.

d_1	**bcGRU**	**HiGRU**	**HiGRU-f**	**HiGRU-sf**
-	65.6(1.2)	-	-	-
300	-	67.2(0.6)	68.4(1.0)	68.9(1.5)
200	-	67.6(2.0)	**68.9**(0.9)	69.1(1.3)
150	-	**67.6**(1.5)	68.5(1.3)	68.9(1.2)
100	-	67.5(1.7)	68.4(1.3)	**69.6**(1.0)

condition can be made between our HiGRU-sf and bcGRU if we set d_1 to 150. From the testing results on Friends in Table 3.5, we can observe that: (1) Under the fair condition, the performance of our HiGRU-sf is not degraded compared to that when $d_1 = 300$. HiGRU-sf still outperforms bcGRU by a significant margin. (2) Overall, no matter d_1 is larger or smaller than 150, HiGRU-sf maintains consistently good performance and the difference between HiGRU-sf and HiGRU-f or HiGRU keeps noticeable. These results further demonstrate the superiority of our proposed models over the baseline bcGRU and the motivation of developing the two variants based on the vanilla HiGRU.

3.4.2 Successful Cases

We investigate three scenes related to the word “okay” that expresses three distinct emotions. The first two scenes come from the testing set of Friends and the third one from that of IEMOCAP. We report the predictions made by bcGRU and our HiGRU-sf, respectively, in Table 3.6. In *Scene 1*, “okay” with period usually exhibits little emotion and both bcGRU and HiGRU-sf correctly classify it as “Neu". In *Scene 2*, “okay” with “!” expresses strong emotion. However, bcGRU misclassifies it to “Ang" while HiGRU-sf successfully recognizes it as “Joy". Actually, the mistake can be traced back to the first utterance of this scene which is also misclassified as “Ang". This indicates that bcGRU tends to capture the wrong atmosphere within the conversation. As for *Scene 3*, “okay” with period now indicates “Sad" and is correctly recognized by HiGRU-

Table 3.6: "Okay" expresses distinct emotions in three different scenes.

Speaker	Utterance	Truth	bcGRU	HiGRU-sf
Scene 1				
Phoebe	Okay. Oh but don't tell them Monica's pregnant because they frown on that.	Neu	Neu	Neu
Rachel	Okay.	Neu	Neu	Neu
Phoebe	Okay.	Neu	Neu	Neu
Scene 2				
Phoebe	Yeah! Sure! Yep! Oh, y'know what? If I heard a shot right now, I'd throw my body on you.	Joy	Ang	Joy
Gary	Oh yeah? Well maybe you and I should take a walk through a bad neighborhood.	Other	/	/
Phoebe	Okay!	Joy	Ang	Joy
Gary	All right.	Neu	Neu	Neu
Scene 3				
Female	Can I send you, like videos and stuff? What about when they start walking.	Other	/	/
Male	Yeah yeah yeah.	Sad	Hap	Sad
Male	You you record every second. You record every second because I want to see it all. Okay?	Hap	Hap	Sad
Male	If I don't get to see it now, I get to see it later at least, you know? You've got to keep it all for me; all right?	Other	/	/
Female	Okay.	Sad	Neu	Sad

sf but misclassified as "Neu" by bcGRU. Note that HiGRU-sf also classifies the third utterance in *Scene 3* as "Sad" which seems to be conflicting to the ground truth. In fact, our HiGRU-sf captures the

Table 3.7: Wrong predictions made by both bcGRU and our HiGRU-sf in two scenes.

Speaker	Utterance	Truth	bcGRU	HiGRU-sf
Scene 4				
Ross	Hi.	Neu	Neu	Neu
Rachel	Hi.	Neu	Neu	Neu
Ross	Guess what?	Neu	Neu	Neu
Rachel	What?	Neu	Neu	Neu
Ross	They published my paper.	Joy	Sad	Neu
Rachel	Oh, really, let me see, let me see.	Joy	Neu	Neu
Phoebe	Rach, look! Oh, hi! Where is my strong Ross Skywalker to come rescue me. There he is.	Other	/	/
Scene 5				
Speaker-1	Sorry for keeping you up	Sad	Sad	Sad
Speaker-2	Lol don't be	Joy	Joy	Joy
Speaker-2	I didn't have to get up today	Neu	Sad	Sad
Speaker-1	:p	Joy	Joy	Joy
Speaker-2	It's actually been a really lax day	Joy	Neu	Sad

blues of this parting situation, where the true label "Hap" may not be that suitable. These results show that our HiGRU-sf learns from both each utterance and the context, and can make correct predictions of the emotion of each utterance.

3.4.3 Failed Cases

At last, we show some examples that both bcGRU and our HiGRU-sf fail in recognizing the right emotions in Table 3.7, i.e., *Scene 4* from Friends and *Scene 5* from EmotionPush. In *Scene 4*, both bcGRU and HiGRU-sf make wrong predictions for the fifth and the sixth utterances. It should be good news that Ross has his paper published and Rachel is glad to see related reports about it. While the repeating of "let me see" two times is actually a good indicator of a non-neutral

emotion for human, it is still hard for the models to understand the meaning of such a behavior. This kind of scenes may be addressed by incorporating some other features like audio and video. As for *Scene 5*, the third and the fifth utterances are classified into wrong emotions. Notice that the emotions indicated from the two utterances are very subtle even for humans. Speaker-2 did not plan to get up today, but Speaker-1 kept him/her up and it ended up with a really lax day. Thus, Speaker-2 feels joyful now. This indicates that even taking into the context into account, the models' capability of understanding subtle emotions is still limited and more exploration is required.

3.5 Summary

In this chapter, we study the intra-sample structure for context enhancement to better address the ERC task. Specifically, we propose a Hierarchical Gated Recurrent Unit (HiGRU) framework to learn both the context of words and the context of utterances. We further promote the HiGRU framework to two variants, HiGRU-f, and HiGRU-sf, and effectively capture the word- and utterance-level inputs and the long-range contextual information, respectively. Experimental results demonstrate that our proposed HiGRU models can sufficiently capture the context information, yielding a significant boost of performance on all three tested datasets. In the future, we plan to explore semi-supervised learning methods to address the problem of data scarcity in this task.

□ **End of chapter.**

Chapter 4

Intra-Sample Structure Mining for Self-Supervised Learning

In this chapter, we investigate the intra-sample structure for self-supervised learning. We also conduct the studies on the ERC task for not only the rich structure information of conversations but more importantly, the data scarcity issue, which could be complemented by unlabeled conversation data. Existing datasets for the ERC task contain inadequate conversations, which prevent the models from playing their maximum effect. However, collecting labeled data by human annotation is costly as annotators are required to recognize either obvious or subtle differences between emotions and consider the context in the conversations. To take the advantage of the massive unlabeled conversation data, we propose a conversation completion (ConvCom) task to pre-train a context-dependent encoder (Pre-CODE) on unlabeled conversation data. Essentially, such a pre-training task utilizes the sequential relationship between utterances in each conversation, enabling the learning of representations at both utterance- and conversation-level. By fine-tuning on the labeled data, our Pre-CODE achieves significant improvements of performance on the ERC task, and particularly benefits the prediction of the minority emotion classes.

Table 4.1: Statistics of the datasets for ERC.

Model	Conversation			Utterance		
	Train	Val	Test	Train	Val	Test
IEMOCAP	96	24	31	3,569	721	1,208
Friends	720	80	200	10,561	1,178	2,764
EmotionPush	720	80	200	10,733	1,202	2,807
EmoryNLP	713	99	85	9,934	1,344	1,328
MOSEI*	2,250	300	676	16,331	1,871	4,662

4.1 Problems and Motivation

Recent studies on the ERC task have employed context-dependent models [5, 6, 39, 52] to exploit the inherent hierarchical structure of conversations (i.e., words-to-utterance and utterances-to-conversation). Despite their remarkable success, context-dependent models suffer from the *data scarcity* issue. In the ERC task, annotators are required to recognize either obvious or subtle difference between emotions, and tag the instance with a specific emotion label, such that labeled data with human annotations are very costly to collect. In addition, existing datasets for ERC [78, 79, 80, 81, 82] contain inadequate conversations, as shown in Table 4.1, which prevent the context-dependent models from playing their maximum effect.

In this chapter, we hope to take advantage of unlabeled conversation data by exploiting the intra-sample structure for self-supervised learning. Specifically, we propose a conversation completion (ConvCom) task based on unlabeled conversation data, which attempts to select the correct answer from candidate answers to fill a masked utterance in a conversation. Then, on the proposed ConvCom task, we Pre-train a basic COntext-Dependent Encoder (Pre-CODE). The hierarchical structure of the context-dependent encoder (CODE) makes our work different from those that focus on universal sentence encoders [24, 26, 8]. Finally, we fine-tune the Pre-CODE on *five* datasets of the ERC task. Experimental results show that the fine-tuned Pre-CODE achieves a significant improvement of performance

over the baselines, particularly on minority emotion classes, demonstrating the effectiveness of our approach. Our contributions of this work are as follows:

- We propose the conversation completion task for the context-dependent encoder to learn from unlabeled conversation data.
- We fine-tune the pre-trained context-dependent encoder on the datasets of ERC and achieve significant improvement of performance over the baselines.
- Extensive analysis suggests that both utterance and conversation encoders are well pre-trained and pre-training particularly benefits the prediction of minority classes.

4.2 Methodology

4.2.1 Pre-training Task

We exploit the self-supervision signal in conversations to construct our pre-training task. Formally, given a conversation D_i from the whole dataset $\mathcal{D}$, denoted as $D_i = \{u_1, u_2, \cdots, u_{N_i}\}$, we mask a target utterance u_j as $D_i \backslash u_j = \{\cdots, u_{j-1}, [mask], u_{j+1}, \cdots\}$ to create a question, and try to retrieve the correct utterance u_j from the whole training corpus. The choice of filling the mask involves countless possible utterances, making it infeasible to formulate the task into a multi-label classification task with softmax. We instead simplify the task into a response selection task [93] using negative sampling [22], which is a variant of noise-contrastive estimation (NCE) [94]. To achieve so, we sample $N-1$ noise utterances elsewhere, along with the target utterance, to form a set of N candidate answers. Then the goal is to select the correct answer, i.e., u_j, from the candidate answers to fill the mask, conditioned on the context utterances. We term this task "Conversation Completion", abbreviated as **ConvCom**. Figure 4.1 shows an example, where the utterance `u4` is masked out from the original conversation and the candidate answers are composed by `u4` and `two` noise utterances.

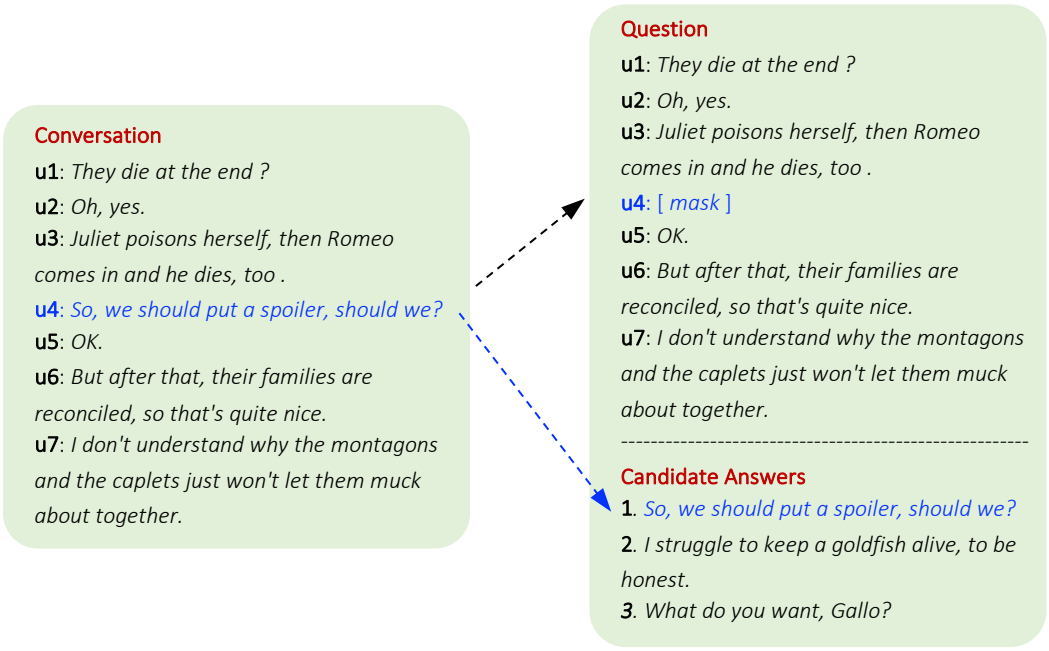


Figure 4.1: A data example in the ConvCom task.

4.2.2 Pre-training Model

Context-Dependent Encoder. The context-dependent encoder consists of two parts: an utterance encoder, and a conversation encoder. For the i^{th} conversation, the j^{th} utterance is represented by a sequence of word vectors $\mathbf{x}_j = \{w_k\}_{k=1}^{M_j}$, where M_j is the number of words. The word vectors are initialized by the 300-dimensional pre-trained GloVe word vectors[1] [23].

For the utterance encoder, we adopt a bidirectional GRU to read the word vectors of an utterance, and produce the hidden state $\overleftrightarrow{\mathbf{h}}_k = [\overrightarrow{\mathbf{h}}_k; \overleftarrow{\mathbf{h}}_k] \in \mathbb{R}^{2d_u}$. We apply *max-pooling* and *mean-pooling* on the hidden states of all words. The pooling results are summed up, followed by a fully-connected layer, to obtain the embedding of the utterance termed $\mathbf{u}_j$:

$$\mathbf{h}_j = \max(\{\overleftrightarrow{\mathbf{h}}_k\}_{k=1}^{M_j}) + \text{mean}(\{\overleftrightarrow{\mathbf{h}}_k\}_{k=1}^{M_j}), \tag{4.1}$$

$$\mathbf{u}_j = \tanh(\mathbf{W}_u \cdot \mathbf{h}_j + \mathbf{b}_u), j \in [1, N_i], \tag{4.2}$$

where N_i is the number of utterances in the i^{th} conversation. In our preliminary experiments, we find that the results variate quite

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

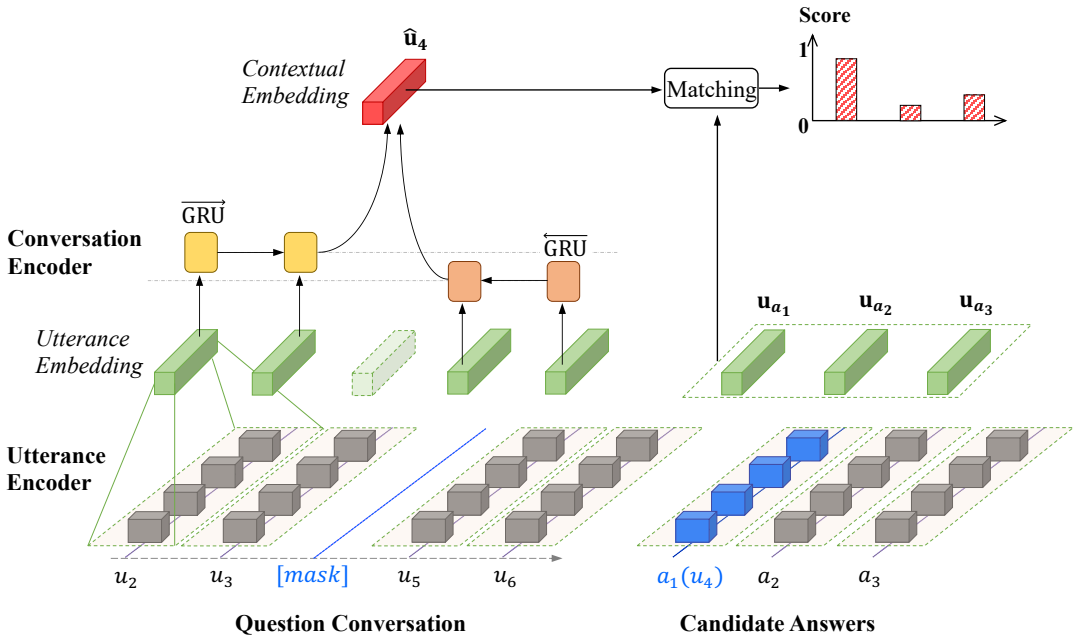


Figure 4.2: The architecture of the context-dependent encoder with the pre-training objective.

significantly using max-pooling though the maximum value could be very high. Therefore, we also apply mean-pooling to add some stable features.

As for the conversation encoder, since an utterance could express different meanings in different contexts, we adopt another bidirectional GRU to model the utterance sequence of a conversation to capture the relationship between utterances. The produced hidden states are termed $\overrightarrow{\mathbf{H}}_j, \overleftarrow{\mathbf{H}}_j \in \mathbb{R}^{d_c}$. With the bidirectional GRU, we are able to capture the context of each utterance from the neighbor utterances before and after it.

Pre-training Objective. To train the context-dependent encoder on the proposed ConvCom task, we construct a contextual embedding for each masked utterance by combining its context from the history $\overrightarrow{\mathbf{H}}_{j-1}$ and the future $\overleftarrow{\mathbf{H}}_{j+1}$ (see Figure 4.2):

$$\hat{\mathbf{u}}_j = \tanh(\mathbf{W}_c \cdot [\overrightarrow{\mathbf{H}}_{j-1}; \overleftarrow{\mathbf{H}}_{j+1}] + \mathbf{b}_c). \tag{4.3}$$

Then, the contextual embedding $\hat{\mathbf{u}}_j$ is matched to the candidate answers to find the most suitable one to fill the mask. To compute

the matching score, we adopt dot-product with a sigmoid function as:

$$s(\hat{\mathbf{u}}_j, \mathbf{u}_{a_n}) = \sigma(\hat{\mathbf{u}}_j^\top \mathbf{u}_{a_n}), n \in [1, N], \tag{4.4}$$

where $\sigma(x) = \frac{1}{(1+\exp(-x))} \in (0, 1)$ is the sigmoid function, and $\mathbf{u}_{a_n}$ is the embedding of the nth candidate answer. The goal is to maximize the score of the target utterance and minimize the score of the noise utterances. Thus the loss function becomes:

$$\mathcal{F} = -\sum_j \left[\log \sigma(\hat{\mathbf{u}}_j^\top \mathbf{u}_{a_1}) + \sum_{n=2}^{N} \log \sigma(-\hat{\mathbf{u}}_j^\top \mathbf{u}_{a_n}) \right], \tag{4.5}$$

where a_1 corresponds to the target utterance, and the summation goes over each utterance of all the conversations in the training set.

4.3 Experiment: Pre-training

4.3.1 Unlabeled Conversation Data

Our unlabeled conversation data comes from an open-source database named OpenSubtitle[2] [83], which contains a large amount of subtitles of movies and TV shows. Specifically, we retrieve the English subtitles throughout the year of 2016, including 25466 `.html` files. We extract the text subtitles from all the `.html` files and pre-process them as below:

- For each episode, we remove the first and the last *ten* utterances in case they are instructions but conversations, especially in TV shows;
- We split the conversations in each episode randomly into shorter ones with *5* to *100* utterances, following a uniform distribution;
- A short conversation is removed if over half of its utterances contain less than *eight* words each. This is done to force the conversation to capture more information;
- All the short conversations are randomly split into a training set, a validation set, and a test set, following the ratio of 90:5:5.

[2] http://opus.nlpl.eu/OpenSubtitles-v2018.php

Table 4.2: Statistics of the created datasets for the ConvCom task.

Set	Conversation	Utterance (Avg.)	Word (Avg.)
Train	58360	41.3	10.1
Val	3186	41.0	10.1
Test	3297	40.8	10.1

Table 4.2 lists the statistics of the resulting sets, where #Conversation denotes the number of conversations in a set, Avg. #Utternace is the average number of utterances in a conversation, and Avg. #Word is the average number of tokens in an utterance. In total, there are over 2 million utterances in over 60k conversations, which are at least 100 times more than those datasets for ERC (see Table 4.1).

4.3.2 Noise Utterances

We randomly sample *ten* noise utterances for each utterance in the training set, validation set, and test set. In each set, the utterances of a conversation is masked one by one in turn. Each utterance is paired with *ten* noise utterances sampled from elsewhere within the set, but we share the *ten* noise utterances for the utterances in the same conversation. During training, we can either use the pre-selected noise utterances or sample an arbitrary number of noise utterances dynamically. We use the validation set to choose model parameters, and evaluate the model performance on the test set.

4.3.3 Evaluation

To evaluate the pre-trained model, we adopt the evaluation metric:

$$\mathbf{R}_{N'}@k = \frac{\sum_{i=1}^{k} y_i}{\sum_{i=1}^{N'} y_i}, \tag{4.6}$$

which is the recall of the true positives among k best-matched answers from N' available candidates for the given contextual embedding $\hat{\mathbf{u}}_j$ [84]. The $\mathbf{R}_{N'}@k$ is calculated for each masked utterance and we report the average results of all utterances. The variate y_i represents

the binary label for each candidate, i.e., 1 for the target one and 0 for the noise ones. Here, we report $\mathbf{R}_5$@1, $\mathbf{R}_5$@2, $\mathbf{R}_{11}$@1, and $\mathbf{R}_{11}$@2.

4.3.4 Training Details

We choose Adam [92] as the optimizer with an initial learning rate of 2×10^{-4}, which is decayed with a rate of 0.75 once the validation recall $\mathbf{R}_{11}$@1 stops increasing. We use a dropout rate of 0.5 for the utterance encoder and the conversation encoder, respectively. Gradient clipping with a norm of 5 is also applied to avoid gradient explosion. Each conversation in the training set is regarded as a batch, where each utterance plays the role of target utterance by turns. We randomly sample 10 noise utterances for each conversation during training and validate the model every epoch. The CODE is pre-trained for at most 20 epochs, and early stopping with a patience of 3 is adopted to choose the optimal parameters. Note that, we fix the word embedding layer during pre-training to focus on the utterance encoder and the conversation encoder.

4.3.5 Results

For simplicity, we term the context-dependent encoder CODE. We train CODE on the created dataset in three different capacities, namely, Small, Medium, and Large, corresponding to different hidden sizes of the BiGRUs.

Table 4.3 lists the results on the test set. For the Small CODE, it is able to select the correct answer for 70.8% instances with 5 candidate answers and 56.2% with 11 candidates. The accuracy is considerably higher than random guesses, i.e., 1/5 and 1/11, respectively. By increasing the model capacity to Medium and Large, we further improve the recalls by several points successively. These results demonstrate that CODE is indeed able to capture the structure of conversations and perform well in the proposed ConvCom task.

Table 4.3: Test results of CODE on the ConvCom task in three capacities.

Model	d_u/d_c	$\mathbf{R}_5$@1	$\mathbf{R}_5$@2	$\mathbf{R}_{11}$@1	$\mathbf{R}_{11}$@2
SMALL	150	70.8	88.0	56.2	72.7
MEDIUM	300	73.8	89.7	60.4	76.4
LARGE	450	77.2	91.3	64.2	79.1

4.4 Experiment: Fine-tuning

4.4.1 ERC Architecture

Since the ConvCom task and the downstream ERC task both have conversations as the input, we can directly transfer the pre-trained CODE models, dubbed PRE-CODE, to the ERC task. In this way, the sentential context and the relationship of utterances learned on the unlabeled conversations can directly benefit the training of the ERC task. The only component we need to add is a fully connected (FC) layer upon PRE-CODE, followed by a softmax function to form the new architecture. Figure 4.3 shows the resulting architecture, in which we also concatenate the context-independent utterance embeddings to the contextual ones before fed to the FC.

We adopt a weighted categorical cross-entropy loss function to optimize the model parameters:

$$\mathcal{L} = -\frac{1}{\sum_{i=1}^{L} N_i} \sum_{i=1}^{L} \sum_{j=1}^{N_i} \omega(c_j) \sum_{c=1}^{|\mathcal{C}|} \mathbf{y}_j^c \log_2(\hat{\mathbf{y}}_j^c), \tag{4.7}$$

where $|\mathcal{C}|$ is the number of emotion classes, $\mathbf{y}_j$ is the one-hot vector of the true label, and $\hat{\mathbf{y}}_j$ is the softmax output. The weight $\omega(c)$ is inversely proportional to the ratio of class c in the training set with a power rate of 0.5.

4.4.2 Compared Methods

We mainly compare our PRE-CODE with a number of previous works: bcLSTM [5], CMN [6], SA-BiLSTM [91], CNN-DCNN [88], SCNN [80], HiGRU [39], and our implementation for the follows:

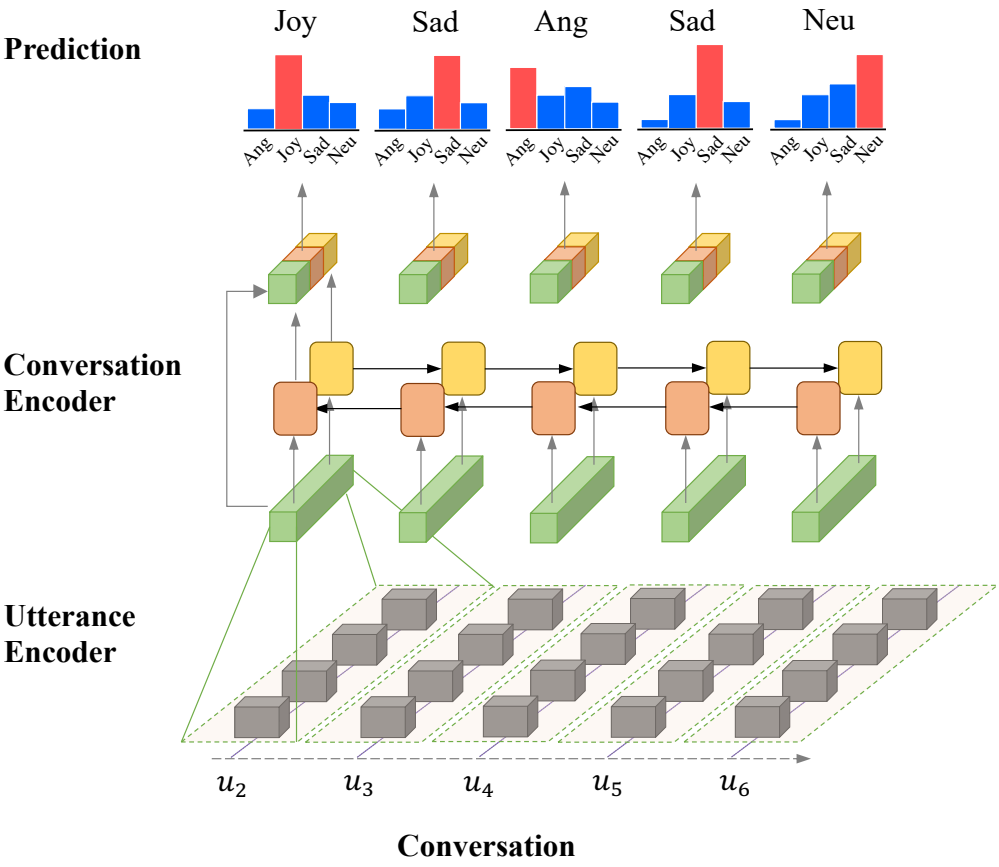


Figure 4.3: The architecture for the ERC task. Both the utterance encoder and conversation encoder are transferred from the PRE-CODE.

- bcLSTM$_*$: bcLSTM re-implemented by us following Jiao et al. [39];
- bcGRU: A variant of bcLSTM$_‡$ implemented with BiGRUs;
- CODE: A vanilla CODE in MEDIUM capacity without pre-training. Unless otherwise stated, the PRE-CODE in the experiments is also in the capacity of MEDIUM by default.

4.4.3 ERC Datasets

We conduct experiments on five datasets for the ERC task, including IEMOCAP[3] [79], Friends[4] [95], EmotionPush[5] [95], EmoryNLP[6] [80],

[3] https://sail.usc.edu/iemocap/
[4] http://doraemon.iis.sinica.edu.tw/emotionlines
[5] http://doraemon.iis.sinica.edu.tw/emotionlines
[6] https://github.com/emorynlp/emotion-detection/

Table 4.4: Test results on IEMOCAP, EmoryNLP, and MOSEI*. The implemented bcLSTM performs much better than the original one, possibly because that the original bcLSTM is not trained end-to-end.

Model	IEMOCAP		EmoryNLP		MOSEI*	
	F1	WA	F1	WA	F1	WA
bcLSTM [5]	–	73.6	–	–	–	–
CMN [6]	–	74.1	–	–	–	–
SCNN [80]	–	–	26.9	**37.9**	–	–
HiGRU-sf [39]	–	82.1	–	–	–	–
bcLSTM*	76.6	77.1	25.5	33.5	29.1	56.3
bcGRU	77.6	78.2	26.1	33.1	28.7	56.4
CODE-MED	78.6	79.6	26.7	34.7	29.7	56.6
PRE-CODE	**81.5**	**82.9**	**29.1**	36.1	**31.7**	**57.1**

and MOSEI[7] [82]. For MOSEI, we pre-process it to adapt to the ERC task and name the pre-processed dataset as MOSEI* here. Specifically, we utilize the raw transcripts of MOSEI, where over 14K utterances are not annotated, and others are labeled with one or more emotion labels. For the unlabeled utterances, we just remove them from the dataset. For the utterance with more than one emotion label, we determine its primary emotion by the majority vote or the highest emotion intensity sum if there are more than one majority vote. For the utterances that obtain zero vote for all emotion classes, we annotate them as *other*.

For the first three datasets, we follow previous works [5, 95] to consider only four emotion classes, i.e., *anger*, *joy*, *sadness*, and *neutral*. We consider all the emotion classes for EmoryNLP as in [80] and six emotion classes (without *neutral*) for MOSEI*. All the datasets contain the training set, validation set, and test set, except for IEMOCAP. Thus, we follow [5] to use the first four sessions of transcripts as the training set, and the last one as the test set. The validation set is extracted from the randomly-shuffled training set with a ratio of 80:20. Please revisit the statistic details of datasets in Table 4.1.

[7] http://immortal.multicomp.cs.cmu.edu/raw_datasets/

Table 4.5: Test results on Friends and EmotionPush.

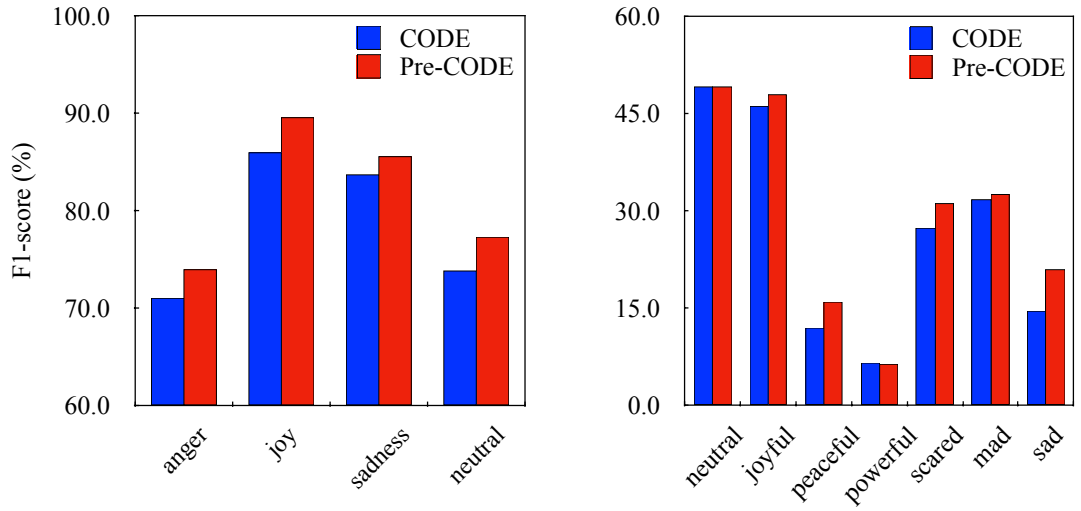
Model	Friends		EmotionPush	
	F1	WA	F1	WA
CNN-DCNN [88];	–	67.0	–	75.7
SA-BiLSTM [91]	–	79.8	–	**87.7**
HiGRU [39]	–	74.4	–	73.8
bcLSTM$_*$	63.1	79.9	60.3	84.8
bcGRU	62.4	77.6	60.5	84.6
CODE-MED	62.4	78.0	60.3	84.2
PRE-CODE	**65.9**	**81.3**	**62.6**	84.7

4.4.4 Evaluation

To evaluate the performance of our models, we report the macro-averaged F1-score [80] and the weighted accuracy (WA) [78] of all emotion classes. The F1-score of each emotion class is also presented for discussion.

4.4.5 Training Details

We still choose Adam as the optimizer and tune the learning rate for the implemented baselines. Generally, the learning rate of 2×10^{-4} works well for all the datasets except MOSEI*, on which we find 5×10^{-5} works better. For the fine-tuning of PRE-CODE, we use the learning rate of the baselines or its half and report the better results here. By default, we fix the word embeddings and tune the pre-trained parameters in PRE-CODE. Fixing all the parameters usually produces limited improvement over the baselines. We monitor the macro-averaged F1-score of the validation set and decay the learning rate once the F1-score stops increasing. The decay rate and patience of early stopping are 0.75 and 6 for all the datasets except IEMOCAP. Since IEMOCAP has much fewer conversations, we change the decay rate and patience of early stopping to 0.95 and 10, respectively.



(a) IEMOCAP (b) EmoryNLP

Figure 4.4: Detailed F1-score of each emotion class on IEMOCAP and EmoryNLP.

4.4.6 Results

We report the main results in Table 4.4 and Table 4.5. Each result is the average of 5 repeated experiments. We directly quote the values of previous studies for specific datasets and run experiments on all the datasets with the implemented bcLSTM$_*$ and bcGRU baselines. As seen, our PRE-CODE outperforms the compared methods on all datasets in terms of F1-score by at least 2.0% absolute improvement. We also conduct significance tests by using two-tailed paired t-tests over the F-1 scores of PRE-CODE and CODE-MED. P-values are obtained as 0.0107, 0.0038, 0.0011, 0.0003, and 0.0068 for IEMOCAP, EmoryNLP, MOSEI*, Friends, and EmotionPush, respectively. Therefore, the result for IEMOCAP is statistically significant with a significance level of 0.05 whereas the other four datasets obtain a significance level of 0.01. It demonstrates the effectiveness of transferring the knowledge from unsupervised conversation data to the ERC task.

To inspect which aspects pre-training helps the most, we present the F1-score of each emotion class on IEMOCAP and EmoryNLP in Figure 4.4. As seen, our PRE-CODE particularly improves the performance on minority emotion classes, e.g., *anger* and *sadness*

in IEMOCAP, and *peaceful* and *sad* in EmoryNLP. These results demonstrate that pre-training can ameliorate the issue of imbalanced performance on minority classes while maintaining good performance on majority classes.

4.5 Analysis

4.5.1 Model Capacity

We investigate how the model performance is affected by the number of parameters, as seen in Table 4.6. We find that: (1) Pre-CODE consistently outperforms CODE in all cases, suggesting that pre-training is an effective method to boost the model performance of ERC regardless of the model capacity. (2) Pre-CODE shows better performance in the capacities of Small and Medium, we speculate that the datasets for ERC are so scarce that they are incapable of transferring the pre-trained parameters of the Large Pre-CODE to optimal ones for ERC.

4.5.2 Layer Effect

We study how different pre-trained layers affect the model performance, as seen in Table 4.7. CODE+Pre-U denotes that only the parameters of utterance encoder are initialized by Pre-CODE. From CODE to CODE+Pre-U and then to Pre-CODE, we conclude that pre-training results in better utterance embeddings and helps the model to capture the utterance-level context more effectively. In addition, Pre-CoDE+Re-W represents that we re-train Pre-CODE for 10 more epochs to adjust the originally fixed word embeddings. The results suggest that pre-training word embeddings do not improve the model performance necessarily but may corrupt the learned utterance and conversation encoders.

Table 4.6: Ablation study on model capacity.

Model	Capacity	IEMOCAP	Friends
CODE	SMALL	76.5	**62.5**
	MEDIUM	**78.6**	62.4
	LARGE	77.6	62.1
PRE-CODE	SMALL	81.2	65.2
	MEDIUM	**81.5**	**65.9**
	LARGE	80.3	64.8

Table 4.7: Ablation study on pre-trained layers.

Layers	IEMOCAP	Friends
PRE-CODE + Re-W	**81.6**	64.5
PRE-CODE	81.5	**65.9**
CODE + Pre-U	80.1	64.8
CODE	78.6	62.4

4.5.3 Qualitative Study

In Table 4.8, we provide two examples for a comparison between CODE and PRE-CODE. The first example is from Friends with consecutive utterances from Joey. It shows that CODE tends to recognize the utterances with exclamation marks "!" as Angry, while those with periods "." as Neutral. The problem also appears on PRE-CODE for short utterances, e.g., "Push!", which contains little and misleading information. This issue might be alleviated by adding other features like audio and video. Still, PRE-CODE performs better than CODE on longer utterances. The other example is from EmotionPush, which are messages with few punctuations. The CODE model predicts almost all utterances as Neutral, which may be because most of the training utterances are Neutral. However, PRE-CODE can identify the minor classes, e.g., Sad, demonstrating that pre-training can alleviate the class imbalance issue.

Table 4.8: Qualitative comparison between CODE and PRE-CODE by two examples.

Speaker	Utterance	Truth	CODE	Pre-CODE
Example 1				
Joey	Come on, Lydia, you can do it.	Neu	Neu	Neu
Joey	Push!	Joy	Ang	Ang
Joey	Push 'em out, push 'em out, harder, harder.	Joy	Neu	Neu
Joey	Push 'em out, push 'em out, way out!	Joy	Ang	Joy
Joey	Let's get that ball and really move, hey, hey, ho, ho.	Joy	Neu	Joy
Joey	Let's... I was just... yeah, right.	Joy	Neu	Neu
Joey	Push!	Joy	Ang	Ang
Joey	Push!	Joy	Ang	Ang
Example 2				
Sp1	It's so hard not to cry	Sad	Ang	Sad
Sp2	What happened	Neu	Neu	Neu
Sp1	I lost another 3 set game	Sad	Neu	Sad
Sp2	It's ok person_145	Neu	Neu	Neu
Sp1	Why does it hurt so much	Sad	Neu	Sad
Sp2	Everybody loses	Neu	Neu	Neu

4.6 Summary

In this chapter, we investigate the intra-sample structure for self-supervised learning on unlabeled conversation data. The proposed conversation completion task is effective for the pre-training of the context-dependent model, which is further fine-tuned to boost the performance of the ERC task significantly. Future directions include exploring advanced models (e.g., TRANSFORMER) for pre-training, conducting domain matching for the unlabeled data, as well as multi-

task learning to alleviate the possible catastrophic forgetting issue in transfer learning.

□ **End of chapter.**

Chapter 5

Inter-Sample Quality Mining for Uncertainty-Based Data Rejuvenation

In this chapter, we study the inter-sample quality for uncertainty-based data rejuvenation. We conduct the studies on the machine translation (MT) task due to its large-scale datasets. Koehn et al. [96] have shown that MT requires a large amount of data to train a well-performing model, especially for neural machine translation (NMT). However, with a rapid increase of data scale, the complex patterns and potential noises in the large-scale data make it more challenging to train NMT models. Therefore, we propose to identify inactive samples in the large-scale datasets and try to re-use them to further boost the performance of NMT models. Specifically, we identify the inactive samples by data uncertainty, which is computed as the output probability of a pre-trained NMT model for each sample. We empirically demonstrate that the existence of inactive samples mainly depends on the data distribution. Further, we introduce data rejuvenation (DataReju) to re-label the inactive samples by forward-translation, and re-use the rejuvenated samples together with active samples to train NMT models. Experimental results on large-scale MT datasets show that our DataReju approach consistently and significantly improves performance for several strong MT models. Extensive analyses reveal that our approach stabilizes and accelerates

the training process of MT models, resulting in final models with better generalization capability.

5.1 Problems and Motivation

Neural machine translation (NMT) is a data-hungry approach, which requires a large amount of data to train a well-performing NMT model [96]. However, the complex patterns and potential noises in the large-scale datasets make training NMT models difficult. While noises in small-scale datasets should also bring negative effects, they can be addressed by human correction, which however is costly and infeasible for large-scale datasets. To relieve this problem, several approaches have been proposed to better exploit the training data, such as curriculum learning [10], data diversification [77], and data denoising [64].

In this chapter, we aim to exploit the inter-sample quality for uncertainty-based data rejuvenation to improve the training of NMT models. Specifically, we explore an interesting alternative which is to reactivate the *inactive samples* in the training data for NMT models. We provide the definition of inactive sample as below:

Definition 3 (Inactive Sample) *Inactive samples are the training samples that only marginally contribute to or even inversely harm the performance of NMT models. According to our following analysis, inactive samples tend to contain noises, rare words, and inconsistent expressions (e.g., passive voice v.s. active voice) or styles (e.g., human translation v.s. natural text) at source and target sides.*

Concretely, we use sentence-level output probability [97] assigned by a trained NMT model to measure the activeness level of training samples, and regard the samples with the least probabilities as inactive samples. Experimental results show that removing 10% most inactive samples can marginally improve translation performance. In addition, we observe a high overlapping ratio (e.g., around 80%) of the most inactive and active samples across random seeds, model capacity, and

model architectures. These results provide empirical support for our hypothesis of the existence of inactive samples in large-scale datasets, which is invariant to specific NMT models and mainly depends on the data distribution itself.

We further propose *data rejuvenation* to rejuvenate the inactive samples so as to improve the performance of NMT models. Specifically, we train an NMT model on the active samples as the rejuvenation model to re-label the inactive samples, resulting in the rejuvenated samples. The final NMT model is trained on the combination of the active samples and rejuvenated samples. Experimental results show that the data rejuvenation approach consistently and significantly improves performance on SOTA NMT models (e.g., LSTM [98], TRANSFORMER [15], and DYNAMICCONV [99]) on the benchmark WMT14 English-German and English-French datasets (sec:main). Encouragingly, our approach is also complementary to existing data manipulation methods (e.g., data diversification [77] and data denoising [64]), and combining them can further improve performance.

Finally, we conduct extensive analyses to better understand the inactive samples and the proposed data rejuvenation approach. Quantitative analyses reveal that the inactive samples are more difficult to learn than the active ones, and rejuvenation can reduce the learning difficulty (§5.4.1). The rejuvenated samples stabilize and accelerate the training process of NMT models (§5.4.2), resulting in final models with better generalization capability (§5.4.3).

Our contributions of this work are as follows:

- We demonstrate the existence of inactive samples in large-scale translation datasets, which mainly depend on the data distribution.
- We propose a general framework to rejuvenate the inactive samples and achieve significant improvements over SOTA NMT models on WMT14 En-De and En-Fr translation tasks, without model modification.

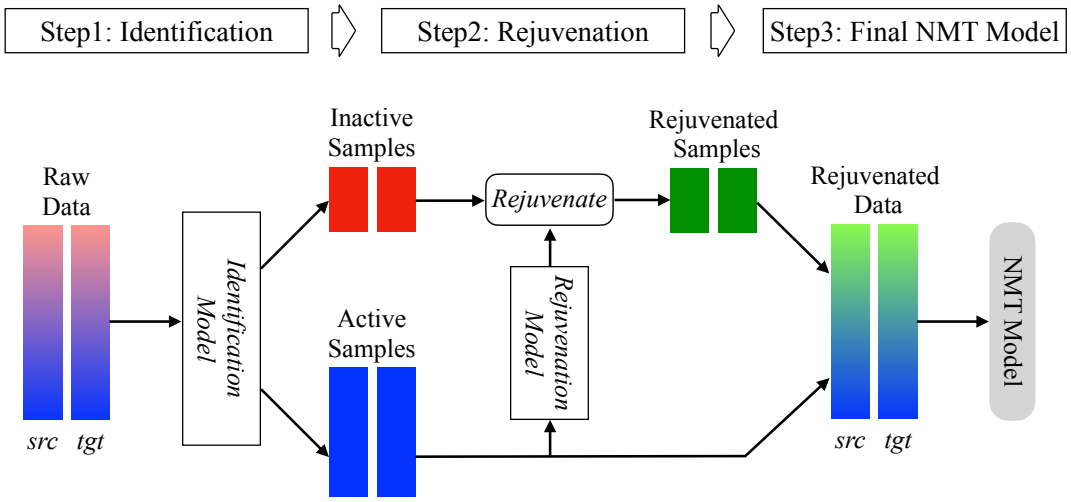


Figure 5.1: The framework of data rejuvenation. The inactive samples from the original training data are identified by the *identification model*, then rejuvenated by the *rejuvenation model*. The rejuvenated samples along with the active samples are used together to train the NMT model.

- We conduct extensive analyses on the linguistic properties of inactive samples, the relationship between human translation and inactive samples, and the effect on the training process before and after rejuvenating inactive samples.

5.2 Methodology

Figure 5.1 shows the framework of the *data rejuvenation* approach, in which we introduce two models: an identification model and a rejuvenation model. The *identification model* distinguishes the inactive samples from the active ones. The *rejuvenation model*, which is trained on the active samples, rejuvenates the inactive samples. The rejuvenated samples and the active samples are combined to train the final NMT model.

There are many possible ways to implement the general idea of data rejuvenation. The aim of this work is not to explore this whole space but simply to show that one fairly straightforward implementation works well and that data rejuvenation helps.

5.2.1 Identification Model

We describe a simple heuristic to implement the identification model by leveraging the output probabilities of NMT models. The training objective of the NMT model is to maximize the log-likelihood of the training data$\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^N$:

$$L(\theta) = \sum_{n=1}^{N} \log P(\mathbf{y}^n|\mathbf{x}^n). \tag{5.1}$$

The trained NMT model assigns a sentence-level probability $P(\mathbf{y}|\mathbf{x})$ to each sentence pair $(\mathbf{x}, \mathbf{y})$, indicating the confidence of the model to generate the target sentence $\mathbf{y}$ from the source one $\mathbf{x}$ [97, 100, 101]. Intuitively, if a training sample has a low sentence-level probability, it is less likely to provide useful information for improving model performance, and thus is regarded as an inactive sample.

Therefore, we adopt sentence-level probability $P(\mathbf{y}|\mathbf{x})$ as the metric to measure the activeness level of each training sample:

$$I(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|\mathbf{x}, \mathbf{y}_{<t}), \tag{5.2}$$

where T is the number of target words in the training sample. $I(\mathbf{y}|\mathbf{x})$ is normalized by the length of target sentence $\mathbf{y}$ to avoid length bias. We train an NMT model on the original training data and use it to score each training sample. We treat a certain percent of training samples with the least sentence-level probabilities as inactive samples. Though there might be other ways to identify the inactive samples, we find the confidence of models adopted here performs well.

5.2.2 Rejuvenation Model

Inspired by recent successes on data augmentation for NMT, we adopt the widely-used back-translation [70] and forward-translation [75] approaches to implement the rejuvenation model. After the active samples are distinguished from the training data, we use them to train an NMT model in the forward direction for forward-translation or/and the reverse direction for back-translation. The trained model

rejuvenates each inactive sample by producing a synthetic-parallel sample based on their source (for forward-translation) or target (for back-translation) side. Benefiting from the knowledge distillation based on active samples, the rejuvenated samples consist of simpler patterns than the original samples [72], thus are more likely to be learned by NMT models.

5.3 Experiment

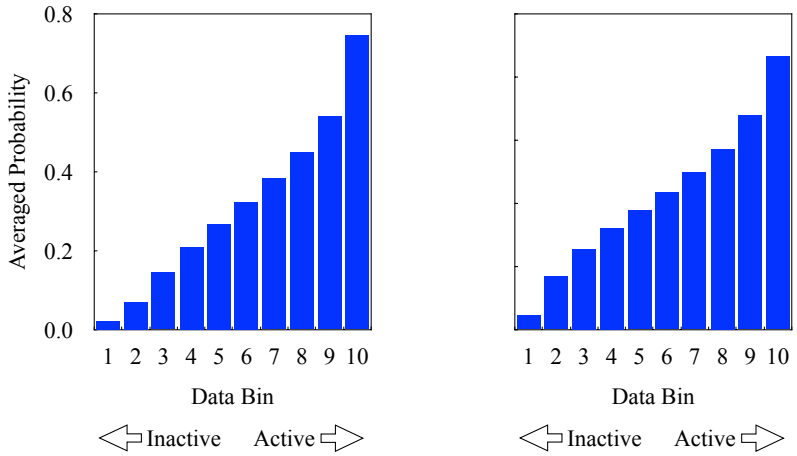
In this section, we conduct experiments to evaluate the effectiveness of our framework. We first introduce the setup of our experiments (§5.3.1), then the ablation studies of the identification model (§5.3.2) and the rejuvenation model (§5.3.3). At last, we report the main results across language pairs from representative NMT architectures (§5.3.4) to demonstrate the universality of our approach.

5.3.1 Experimental Setup

Data. We conduct experiments on the two benchmark datasets, i.e., WMT14 English⇒German (En⇒De) and English⇒French (En⇒Fr), which consist of about 4.5M and 35.5M sentence pairs, respectively. We apply BPE [102] with 32K merge operations for both language pairs. BPE is used to segment text into subword units, which can highly reduce the size of vocabulary and improve the learning of low-frequency words. The experimental results are reported in case-sensitive BLEU score [85] on the test sets.

Model. We validate our approach on a couple of representative NMT architectures:

- LSTM [98] that is implemented in the TRANSFORMER framework.
- TRANSFORMER [15] that is based solely on attention mechanisms.
- DYNAMICCONV [99] that is implemented with lightweight and dynamic convolutions, which can perform competitively to the best reported TRANSFORMER results.



(a) En⇒De (b) En⇒Fr

Figure 5.2: Probability diagram on (a) En⇒De and (b) En⇒Fr datasets. Training samples in smaller bins (e.g., 1, 2) are regarded as inactive samples due to their lower probabilities.

We adopt the open-source toolkit Fairseq [103] to implement the above NMT models. We follow the settings in the original works to train the models. In brief, we train the LSTM model for 100K steps with 32K (4096×8) tokens per batch. For TRANSFORMER, we train 100K and 300K steps with 32K tokens per batch for the BASE and BIG models respectively. We train the DYNAMICCONV model for 30K steps with 459K (3584×128) tokens per batch. We select the model with the best perplexity on the validation set as the final model.

We first conduct ablation studies on the identification model (§5.3.2) and rejuvenation model (§5.3.3) on the WMT14 En⇒De dataset with TRANSFORMER-BASE. Then we report the translation performance on different model architectures and language pairs, as well as the comparison with previous studies (§5.3.4).

5.3.2 Identification of Inactive Samples

In this section, we investigate the reasonableness and consistency of the identified inactive samples.

Identified Inactive Samples. As aforementioned, we rank the training samples according to the sentence-level output probability

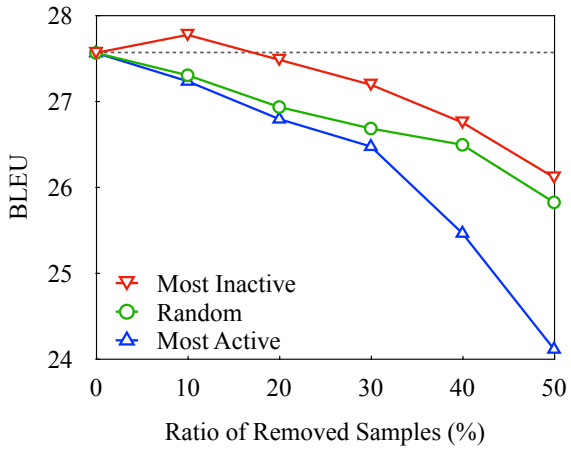


Figure 5.3: Translation performance of the NMT model trained on the training data with the most inactive samples removed. For comparison, results of the most active samples and randomly sampled samples are also presented.

(i.e., confidence) assigned by a trained NMT model. We follow Wang et al. [100] to partition the training samples into 10 equal bins (i.e., each bin contains 10% of training samples) according to the ranking of their probabilities and report the averaged probability of each bin, as depicted in Figure 5.2. The distributions of En⇒De and En⇒Fr are similar due to the ranking of both training sets. As seen, the samples in the 1^{st} data bin have much lower probabilities than the other ones and those in the 10^{th} data bin have much higher probabilities. In contrast, the probabilities from the 2^{nd} to the 9^{th} data bins increases linearly. It suggests that the NMT models tend to give special priority to the easiest samples and hardly learn the most difficult ones. Accordingly, we treat the samples in the 1^{st} data bin as inactive samples.

Reasonableness of Identified Inactive Samples. In this experiment, we evaluate the reasonableness of the identified inactive samples by measuring their contribution to the translation performance. Intuitively, a reasonable set of inactive samples can be removed from the training data without harming the translation performance, since they cannot provide useful information to the NMT models. Starting from this intuition, we remove a certain percentage of samples with the least probabilities (e.g., most inactive samples) from the training data, and evaluate the performance of the NMT model that is trained

on the remaining data.

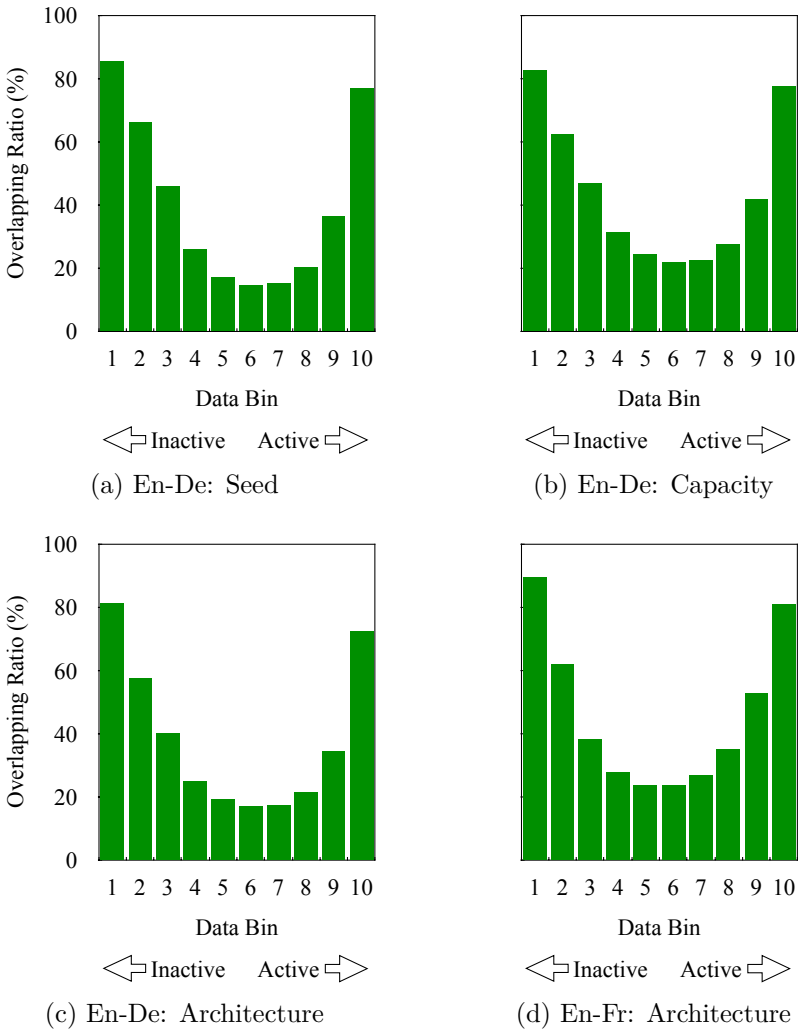
Figure 5.3 shows the contribution of the most inactive samples to translation performance. Generally, the performance drop grows up with the increased portion of samples being removed from the training data. The declining trend of the inactive samples is more gentle than the randomly selected samples, and that of the active samples is steepest. These results demonstrate the reasonableness of the identified samples. Encouragingly, the translation performance does not degrade when removing 10% of the most inactive samples, which is consistent with the finding of Birodkar et al. [66] on the CV datasets.

Consistency of Identified Inactive Samples. Since our identification of inactive samples relies on a pre-trained NMT model, one doubt naturally arises: *Are the identified inactive samples model-specific*? For example, different NMT models treat different portions of the training data as inactive samples. To dispel the doubt, we identify some factors that can significantly affect the performance of NMT models:

- *Random Seed*: We use five different seeds for Transformer-Base: "1", "12", "123", "1234", and "12345".
- *Model Capacity*: We vary the number of layers and layer dimensionality of Transformer: Tiny (3×256), Base (6×512), and Big (6×1024).
- *Model Architecture*: We use the aforementioned architectures: Lstm, Transformer-Base, and DynamicConv.

For each data bin, we calculate the ratio of samples that are shared by different model variants (e.g., different random seeds). Generally, a high overlapping ratio denotes the identified samples are more agreed by different models, which suggests the samples are not model-specific.

Figure 5.4 depicts the results. As expected, there is always a high overlapping ratio (over 80%) for the most inactive samples (i.e., 1^{st} data bin) across model variants and language pairs. The



(a) En-De: Seed (b) En-De: Capacity

(c) En-De: Architecture (d) En-Fr: Architecture

Figure 5.4: Ratio of samples that are shared by different model variants: random seed (a), model capacity (b), model architecture on En⇒De (c) and En⇒Fr (d) datasets. A high overlapping ratio for most inactive samples (i.e., 1^{st} data bin) demonstrates that the identified inactive samples are not model-specific.

high consistency of identified inactive samples demonstrates that *the proposed identification is invariant to specific models, and depends on the data distribution itself.* Another interesting finding is that the most active samples (i.e., 10^{th} data bin) also hold a high agreement by model variants. The overlapping ratios of all model variants (i.e., seeds, capacities, and architectures, 9 models in total) on the En⇒De dataset are 70.9%, and 62.5% for the most inactive and (most) active samples, respectively. This indicates that deep learning methods share a common ability to learn from the training samples.

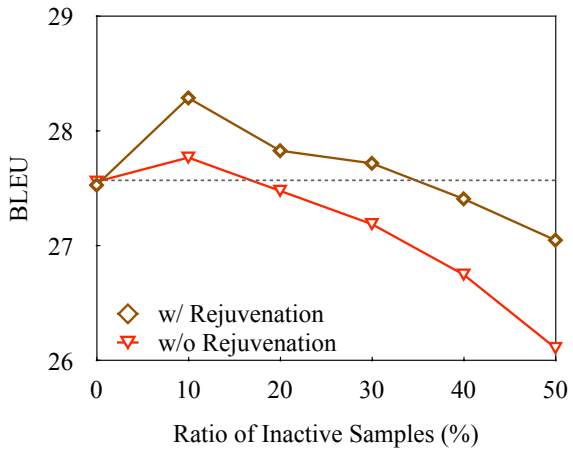


Figure 5.5: Effect of the ratio of samples labeled as inactive samples. We used forward-translation as the rejuvenation strategy and trained the final NMT model on the combination of rejuvenated samples and active samples from scratch.

5.3.3 Rejuvenation of Inactive Samples

In this section, we evaluate the impact of different components on the rejuvenation model.

Ratio of Samples Labeled as Inactive. After all samples are assigned a sentence-level probability by the identification model, we label $R\%$ of samples with the least probabilities as the inactive samples. We investigate the effect of different R on translation performance, as shown in Figure 5.5. Clearly, rejuvenating the inactive samples consistently outperforms its non-rejuvenated counterpart, demonstrating the necessity of the data rejuvenation. Concerning the rejuvenation model, the BLEU score decreases with the increase of R. This is intuitive, since samples with relatively higher probabilities (e.g., beyond the 10% most inactive samples) can provide useful information for NMT models, and rejuvenating them would inversely harm the translation performance. In the following experiments, we treat 10% samples with the least probabilities as inactive samples.

Effect of Rejuvenation Strategy. Table 5.1 lists the results of different rejuvenation strategies. Surprisingly, the back-translation strategy does not improve performance. One possible reason is that the inactive samples are identified by a forward-translation model

Table 5.1: Effect of different rejuvenation strategies.

Rejuvenation	BLEU	△
n/a	27.5	–
Forward Translation	**28.3**	+0.8
Back-Translation	27.5	+0.0
Both	27.8	+0.3

(§5.2.1), indicating that these inactive samples are more difficult for NMT models to generate from the source side to the target side, rather than in the reverse direction. We conjecture that the forward translation strategy may alleviate this problem by constructing a synthetic sample, in which each source side is associated with a simpler target side. Combining both strategies cannot further improve translation performance. In the following experiments, we use forward translation as the default rejuvenation strategy.

Benefiting from Forward Translation or Data Rejuvenation? Some researchers may doubt: *does the improvement indeed come from data rejuvenation, or just from forward translation?* To dispel the doubt, we conduct the comparison experiment by randomly selecting 10% training samples as inactive samples and applying data rejuvenation with the forward translation strategy. As shown in Table 5.2, removing 10% random samples inversely harms the translation performance, and rejuvenating them leads to a further decrease of performance. In contrast, the proposed data rejuvenation improves performance as expected. These results provide empirical support for our claim that the improvement comes from the proposed data rejuvenation rather than forward translation.

5.3.4 Main Results

Comparison with Vanilla Models. Table 5.3 lists the results across model architectures and language pairs. "+ Data Rejuvenation" represents the model trained with our rejuvenated data. Our implemented TRANSFORMER models have more parameters than previous work [15] since we use separate word embeddings for the

Table 5.2: Comparing data rejuvenation on identified inactive samples and forward translation on randomly sampling samples.

Training Data	**BLEU**	△
Raw Data	27.5	–
- 10% *Inactive* Samples	27.8	+0.3
+ Rejuvenated Samples	28.3	+0.8
- 10% *Random* Samples	27.4	-0.1
+ Rejuvenated Samples	27.3	-0.2

input and output of the decoder, which leads to better performance, especially on the large-scale En⇒Fr dataset (e.g., more than 1 BLEU point). Our TRANSFORMER models achieve better results than that reported in previous work [15], especially on the large-scale En⇒Fr dataset (e.g., more than 1.0 BLEU points). Ott et al. [16] show that models of larger capacity benefit from training with large batches. Analogous to DYNAMICCONV, we train another TRANSFORMER-BIG model with 459K tokens per batch ("+ Large Batch" in Table 5.3) as a strong baseline. We test statistical significance with paired bootstrap resampling [104] using `compare-mt`[1] [105] with 1000 re-samples.

Clearly, our data rejuvenation consistently and significantly improves translation performance in all cases, demonstrating the effectiveness and universality of the proposed data rejuvenation approach. It's worth noting that our approach achieves significant improvements without introducing any additional data and model modification. It makes the approach robustly applicable to most existing NMT systems.

Comparison with Previous Work. The proposed *data rejuvenation* approach belongs to the family of data manipulation. Accordingly, we compare it with several widely-used manipulation strategies: data diversification [77], and data denoising [64].

For data diversification, we use both forward-translation [FT, 75] and back-translation [BT, 70] strategies on the original training data,

[1] https://github.com/neulab/compare-mt

Table 5.3: Evaluation of translation performance across model architectures and language pairs. "↑ / ⇑": indicate statistically significant improvement over the corresponding baseline $p < 0.05/0.01$ respectively.

System	Architecture	En⇒De		En⇒Fr	
		BLEU	△	BLEU	△
Existing NMT Systems					
Vaswani et al. [15]	Transformer-Base	27.3	–	38.1	–
	Transformer-Big	28.4	–	41.0	–
Ott et al. [16]	Scale Transformer	29.3	–	43.2	–
Wu et al. [99]	DynamicConv	29.7	–	43.2	–
Our NMT Systems					
This work	Lstm	26.5	–	40.6	–
	+ Data Rejuvenation	27.0$^{\uparrow}$	+0.5	41.1$^{\uparrow}$	+0.5
	Transformer-Base	27.5	–	40.2	–
	+ Data Rejuvenation	28.3$^{\Uparrow}$	+0.8	41.0$^{\Uparrow}$	+0.8
	Transformer-Big	28.4	–	42.4	–
	+ Data Rejuvenation	29.2$^{\Uparrow}$	+0.8	43.0$^{\uparrow}$	+0.6
	+ Large Batch	29.6	–	43.5	–
	+ Data Rejuvenation	30.3$^{\Uparrow}$	+0.7	44.0$^{\uparrow}$	+0.5
	DynamicConv	29.7	–	43.3	–
	+ Data Rejuvenation	30.2$^{\uparrow}$	+0.5	43.9$^{\uparrow}$	+0.6

Table 5.4: Comparison with other data manipulation approaches. Results are reported on the En⇒De test set.

Model	BLEU	△
Transformer-Base	27.5	–
+ *Data Rejuvenation*	28.3	+0.8
+ Data Diversification-BT	26.9	-0.6
+ *Data Rejuvenation*	27.9	+0.4
+ Data Diversification-FT	28.1	+0.6
+ *Data Rejuvenation*	28.5	+1.0
+ Data Denoising	28.1	+0.6
+ *Data Rejuvenation*	28.6	+1.1

and no monolingual data is introduced. The final NMT model is trained on the combination of the original and the synthetic parallel data. Our approach is similar to "Data Diversification-FT" except

that we only forward-translate the identified inactive samples (10% of the training data), while they forward-translate all the training samples.

For data denoising, we rank the training data according to a noise metric, which requires a set of trusted samples. Following Wang et al. [64], we use WMT newstest 2010-2011 as the trusted data, which consists of 5492 samples. The trained NMT model on the raw data is regarded as the noisy model, which is then fine-tuned on the trusted data to obtain the denoised model. For each sentence pair, a noise score is computed based on the noisy and denoised models, which is used for instance sampling during training.

Table 5.4 shows the comparison results on the WMT14 En⇒De test set. All approaches improve translation performance individually except for data diversification with back-translation. Our approach can obtain further improvement on top of these manipulation approaches, indicating that data rejuvenation is complementary to them.

In addition, we compute the overlapping ratio between the noisiest and most inactive samples (10% of the training data) identified by data denoising and data rejuvenation approaches, respectively. We find that there are only 32% of samples that are shared by the two approaches, indicating that the inactive samples are not necessarily noisy samples. In order to better understand the characteristics of inactive samples, we will give more detailed analyses on linguistic properties of the inactive samples in Section 5.4.1.

Random Seeds. Some researchers may doubt if the improvement achieved by our approach comes from lucky random starts. To dispel this doubt, we conduct experiments on the En⇒De dataset using the Transformer-Base model with three random seeds (i.e., 1, 12, and 123). Our approach consistently outperforms the baseline model in all cases (i.e., 28.3/27.5, 28.2/27.4, and 27.9/27.1), demonstrating the effectiveness of our approach.

Source Language. Some researchers may have questions about

Table 5.5: New testing rule on WMT19 En⇒De datasets, evaluated on newstest2019 and newstest2020.

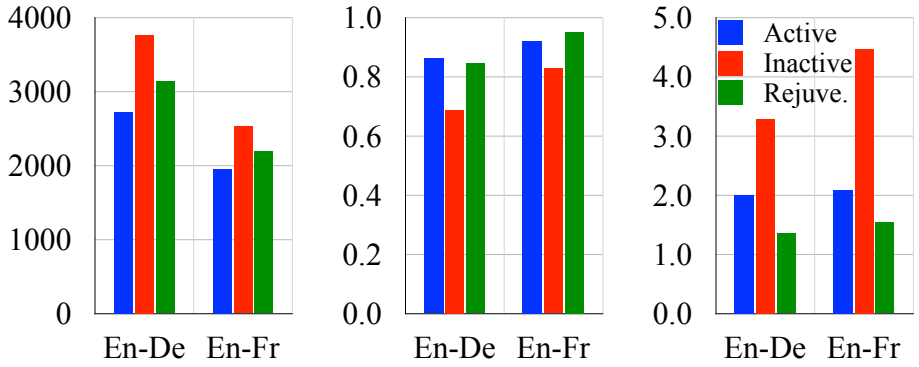
Model	**Newstest2019**		**Newstest2020**	
	BLEU	△	BLEU	△
TRANSFORMER-BIG	41.1	–	33.7	–
+ Data Rejuvenation	43.0	+1.9	35.5	+1.8

the language pairs used in the experiments that both language pairs have English as the source language, which could determine the rejuvenation strategy. To demonstrate the universality of our approach across language directions, we conduct an experiment on the WMT14 De-En translation task. The TRANSFORMER-BASE model achieves a BLEU score of 31.2, and the data rejuvenation approach improves performance by +0.6 BLEU point.

New Testing Setup. Starting from WMT2019 [106], the test sets only include naturally occurring text at the source-side to make a more realistic scenario for practical translation usage. We thus evaluate our approach under this new testing setup. Specifically, we train TRANSFORMER-BIG models on the WMT19 En⇒De datasets with 36.8M sentence pairs and evaluated on newstest2019 and newstest2020. The results are listed in Table 5.5. As seen, our data rejuvenation approach achieves +1.9 and +1.8 improvements of BLEU score on the two test sets, respectively, demonstrating that our approach is even more effective under this new testing setup.

5.4 Analysis

In this section, we perform an extensive study to understand inactive samples and data rejuvenation in terms of linguistic properties (§5.4.1), learning stability (§5.4.2) and generalization capacity (§5.4.3). We also investigate the strategy to speed up the pipeline of data rejuvenation (§5.4.4). Unless otherwise stated, all experiments are conducted on the En⇒De dataset with TRANSFORMER-BASE.



(a) Frequency Rank (b) Coverage (c) Uncertainty

Figure 5.6: Linguistic properties of different training samples: frequency rank (↑ more difficult), coverage (↓ more difficult), and uncertainty (↑ more difficult).

5.4.1 Linguistics Properties

In this section, we investigate the linguistic properties of the identified inactive samples. We explore the following three types of properties:

- *Frequency Rank*: We adopt frequency rank to measure the rarity of words in the target sentences. In the target vocabulary, words are sorted in the descending order of their frequencies in the whole training data, and the frequency rank of a word is its position in the dictionary. Therefore, the higher the frequency rank is, the more rare the word is in the training data. We report the averaged frequency rank of each of the three subsets. The larger frequency rank of inactive samples indicates that they contain more rare words, which makes them more difficult to be learned by NMT models than the active samples.

- *Translation Coverage*: We adopt coverage to measure the ratio of source words being aligned by any target words [107]. Firstly, we train an alignment model on the training data by *fast-align*[2] [108], and force-align the source and target sentences of each subset. Then, we calculate the coverage of each source sentence, and report the averaged coverage of each subset. The lower coverage of inactive samples indicates that they are not well aligned as the active samples, which also makes them more difficult for NMT

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

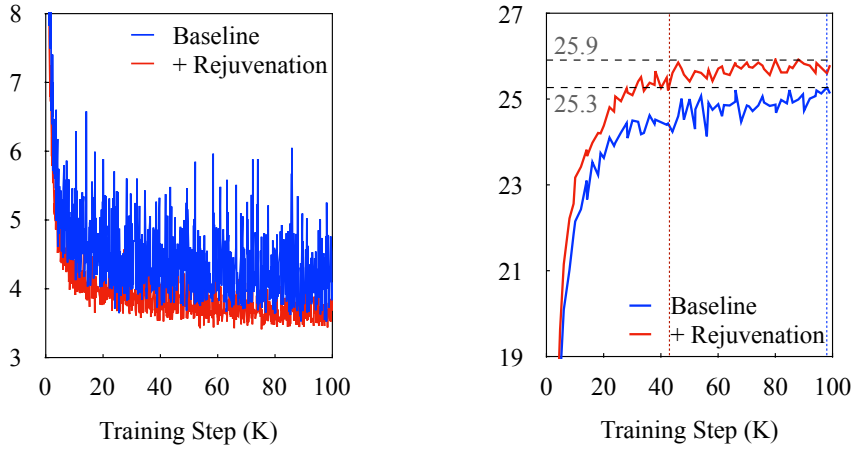
models to learn.

- *Translation Uncertainty*: Uncertainty measures the level of multi-modality of a parallel corpus [109]. We adopt uncertainty to reflect the number of possible translations at the target side for a source sentence. We use the corpus level uncertainty, which measures the complexity of each subset. Corpus level uncertainty is simplified as the sum of the entropy of target words conditioned on the aligned source words denoted $H(y|x = x_t)$. Therefore, an alignment model is also required. To prevent uncertainty from being dominated by frequent words, we followed Zhou et al. [109] to calculate uncertainty by averaging the entropy of target words conditioned on a source word denoted $\frac{1}{|\mathcal{V}_x|}\sum_{x\in\mathcal{V}_x} H(y|x)$. The larger uncertainty of inactive samples indicates that there are more possible translations for each source sentence within. In other words, inactive samples contain more complex patterns, which are more difficult to be learned by NMT models.

Figure 5.6 depicts the results. In summary, the linguistic properties consistently suggest that inactive samples are more difficult than those active ones. By rejuvenation, the inactive samples are transformed into much simpler patterns such that NMT models are able to learn from them. Our findings have also been demonstrated in the non-autoregressive translation (NAT) scenario [110], which usually requires knowledge distillation (i.e., forward-translation) to simplify the training data.

5.4.2 Learning Stability

In this section, we study how data rejuvenation improves translation performance from the perspective of the optimization process, as shown in Figure 5.7. Concerning the training loss (Figure 5.7(a)), our approach converges faster and presents much less fluctuation than the baseline model during the whole training process. The reason could be that the inactive samples are rejuvenated through beam search,



(a) Training Loss (b) Validation BLEU

Figure 5.7: Learning curves on the En⇒De dataset.

which produces high-confidence outputs that correspond to low losses during training. Accordingly, the BLEU score on the validation set is significantly boosted (Figure 5.7(b)). These results suggest that data rejuvenation is able to accelerate and stabilize the training process.

5.4.3 Generalization Capability

In this section, we investigate how data rejuvenation affects the generalization capability of NMT models with two measures, namely, Margin [111] and Gradient Signal-to-Noise Ratio [GSNR, 112]. The two measures are introduced as below:

- *Margin*: Margin [111] is a classic concept in support vector machine, measuring the geometric distance between the support vectors and the decision boundary. To apply margin for NMT models, we followed Li et al. [113] to compute word-wise margin, which is defined as the probability of the correctly predicted word minus the maximum probability of other word types. We computed the word-wise margin over the training set and reported the averaged value.

- *GSNR*: The GSNR metric [112] is proposed to positively correlate with generalization performance. The calculation of a parameter's GSNR is defined as the ratio between its gradient's squared

Table 5.6: Results of generalization capability on the En⇒De dataset. Larger Margin and GSNR values denote better generalization capability.

Model	Margin	GSNR
TRANSFORMER-BASE	0.68	5.2e-3
+ Data Rejuvenation	0.71	8.5e-3

mean and variance over the data distribution. For NMT models, we compute GSNR of each parameter and reported the averaged value over all the parameters.

Table 5.6 lists the results, in which the GSNR values are at the same order of magnitude as that reported by Liu et al. [112]. As seen, our approach achieves noticeably larger Margin and GSNR values, demonstrating that data rejuvenation improves the generalization capability of NMT models.

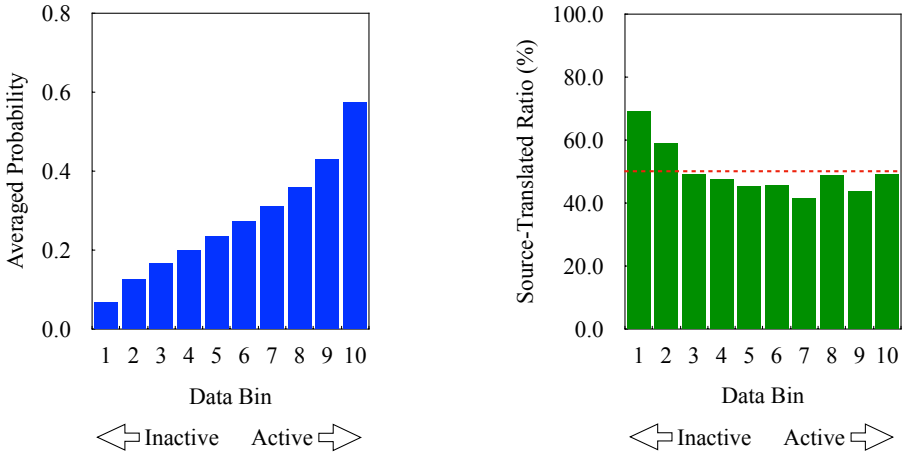
5.4.4 Speeding Up

The pipeline of data rejuvenation in Figure 5.1 is time-consuming: training the identification and rejuvenation models in sequence as well as the scoring and rejuvenating procedures make the time cost of data rejuvenation more than 3X that of the standard NMT system. To save the time cost, a promising strategy is to let the identification model take the responsibility of rejuvenation. Therefore, we use the TRANSFORMER-BIG model with the large batch configuration trained on the raw data to accomplish both identification and rejuvenation. The resulted data is used to train two final models, i.e., TRANSFORMER-BIG and DYNAMICCONV.

Table 5.7 lists the results. With almost no decrease in translation performance, the time cost of data rejuvenation is reduced by about 33%. This makes the total time cost comparable with those data manipulation or augmentation techniques that require additional NMT systems, such as data diversification [77] and back-translation [70]. In addition, the superior performance of DYNAMICCONV (i.e., 30.4) further demonstrates the high agreement of inactive samples across

Table 5.7: Results of speeding up ("Rej.–Big") on the WMT14 En⇒De dataset. "Time" denotes the time of the whole process using 4 NVIDIA Tesla V100 GPUs.

Method	Trans.-Big		Dyn.Conv	
	BLEU	Time	BLEU	Time
Standard	29.6	32h	29.7	31h
Rejuvenate	30.3	+65h	30.2	+62h
Rej.–Big	30.2	+33h	30.4	+32h



(a) Probability (b) Source-Translated Ratio

Figure 5.8: Probability and ratio of source-translated samples over the data bins of En⇒De test set.

architectures.

5.4.5 Inactive Sample Cases

Human Translations from Target to Source as Inactive Samples? Since forward translation performs better than back-translation for rejuvenation, one would wonder if the inactive samples correspond to human translations from target to source. For simplicity, we name such samples as source-translated whereas source-natural otherwise. The information of source-translated/natural samples is unavailable for training samples, but fortunately is provided for test sets[3]. The test set of En⇒De contains 1500 source-translated and 1503 source-natural samples. We split the test samples of En⇒De into 10

[3] https://www.statmt.org/wmt14/test-full.tgz

data bins according to the sentence-level probability (see Eq. (5.2)) of the identification model (i.e., TRANSFORMER-BASE), and then calculate the ratio of source-translated samples in each bin. As seen in Figure 5.8, the ratios of source-translated samples in 1^{st} and 2^{nd} bins (i.e., 69% and 59%) significantly exceed that in the whole test set (i.e., 1500/3003), suggesting that human translations from target to source are more likely to be inactive samples. Such a kind of test samples are also considered inconsistent with the practical testing scenario [106]. From WMT2019, the test samples are all created as human translations from natural source sentences to the target language. Under this new testing setup, Wang et al. [114] suggests that training NMT models on only human translations from the source to the target language can achieve better performance than that on the whole training data. Therefore, we expect our data rejuvenation to be effective mainly on the source-natural samples but possibly not on the source-translated samples.

Case Study. By inspecting the inactive samples, we find that the target sentences tend to be paraphrases of the source sentences rather than direct translations. We provide two cases in Table 5.8. In the first case, the target sentence does not translate "finished the destruction of the first" in the source sentence directly but rephrases it as "tat dann das seine und zerstörte den Rest", meaning "then did his and destroyed the rest" (that was not destroyed by The First World War). As for the second case, "denied by the latter" uses passive voice but its corresponding phrase in the target sentence is in active voice. These observations indicate that the inconsistent structure or expression between source and target sentences could make the samples difficult for NMT models to learn well.

5.5 Summary

In this chapter, we exploit the inter-sample quality for uncertainty-based data rejuvenation so as to improve the training of NMT models. Specifically, we propose data rejuvenation to re-activate the inactive

Table 5.8: Inactive samples from the training sets of En⇒De and En⇒Fr. `X`, `Y` and `Y'` represent the source sentence, target sentence, and the rejuvenated target sentence, respectively. `Y` and `Y'` are also translated into English (`=>En:`) by Google Translate for reference. For either sample, the underlined phrases correspond to the same content.

	Side	Sentence
En⇒De	`X`	The Second World War finished the destruction of the first .
	`Y`	Der zweite Weltkrieg tat dann das seine und zerstörte den Rest . `=>En:` The Second World War then did his and destroyed the rest .
	`Y'`	Der Zweite Weltkrieg beendete die Zerstörung des ersten . `=>En:` The Second World War ended the destruction of the first .
En⇒Fr	`X`	Anything denied by the latter was effectively confirmed as true .
	`Y`	Tout ce que démentait cette agence se révélait dans la pratique bien réel . `=>En:` Everything that this agency denied turned out to be very real in practice .
	`Y'`	Toute chose niée par ce dernier a été effectivement confirmée comme vraie . `=>En:` Anything denied by the latter has actually been confirmed to be true .

training samples for neural machine translation on large-scale datasets. The proposed data rejuvenation scheme is a general framework where one can freely define, for instance, the identification and rejuvenation models. Experimental results on different model architectures and language pairs demonstrate the effectiveness and universality of the data rejuvenation approach. Future directions include exploring advanced identification and rejuvenation models that can better reflect the learning abilities of NMT models, as well as validating on other NLP tasks such as dialogue and summarization.

□ **End of chapter.**

Chapter 6

Inter-Sample Quality Mining for Uncertainty-Based Self-Training Sampling

In this chapter, we investigate the inter-sample quality for uncertainty-based self-training sampling to leverage monolingual (i.e., unlabeled) data in a more efficient way. Self-training augments the training of NMT models with synthetic parallel data, which is usually constructed from a randomly selected subset of the large-scale monolingual data. However, we empirically show that random sampling is sub-optimal and propose to select the most informative monolingual sentences for self-training. To this end, we compute the translation uncertainty of monolingual sentences using the bilingual dictionary extracted from the authentic parallel data. Intuitively, monolingual sentences with lower uncertainty generally correspond to easy-to-translate patterns which may not provide additional gains. Therefore, we propose an uncertainty-based sampling (UncSamp) strategy, which prefers to sample monolingual sentences with relatively higher uncertainty, for self-training. Experimental results on large-scale MT datasets demonstrate that our UncSamp approach improves the translation quality, especially for uncertain sentences, and also the prediction accuracy of low-frequency words at the target side.

6.1 Problems and Motivation

Leveraging large-scale unlabeled data has become an effective approach for improving the performance of natural language processing (NLP) models [8, 26]. As for neural machine translation (NMT), compared to the parallel data, the monolingual data is available in large quantities for many languages. Several approaches on boosting the NMT performance with the monolingual data have been proposed, e.g., data augmentation [70, 75], semi-supervised training [115, 116, 117], pre-training [37, 118, 119]. Among them, data augmentation with the synthetic parallel data [70, 71] is the most widely used approach due to its simple and effective implementation. It has been a de-facto standard in developing the large-scale NMT systems [120, 121, 122].

In this chapter, we exploit the inter-sample quality for uncertainty-based self-training sampling to utilize the large-scale monolingual data more efficiently. Self-training [75] is one of the most commonly used approaches for data augmentation. Generally, self-training is performed in three steps: 1) randomly sample a subset from the large-scale monolingual data; 2) employ a "teacher" NMT model to translate the subset data into the target language to construct the synthetic parallel data; and 3) combine the synthetic and authentic parallel data to train a "student" NMT model. Recent studies have shown that synthetic data manipulation [71, 73] and training strategy optimization [76, 32] in the last two steps can boost the self-training performance significantly. However, how to efficiently and effectively sample the subset from the large-scale monolingual data in the first step has not been well studied.

Intuitively, self-training simplifies the complexity of generated target sentences [123, 109], and easy patterns in monolingual sentences with deterministic translations may not provide additional gains over the self-training "teacher" model [20]. Related work on computer vision also reveals that easy patterns in unlabeled data with the deterministic prediction may not provide additional gains [124]. Therefore,

here we investigate and identify the uncertain monolingual sentences which implicitly hold difficult patterns and exploit them to boost the self-training performance. Specifically, we measure the uncertainty of the monolingual sentences by using a bilingual dictionary extracted from the authentic parallel data. Experimental results show that NMT models benefit more from the monolingual sentences with higher uncertainty, except on those with excessively high uncertainty. By conducting the linguistic property analysis, we find that extremely uncertain sentences contain relatively poor translation outputs, which would hurt the training of NMT models hence the final performance.

Inspired by the above finding, we propose an uncertainty-based sampling strategy for self-training, in which monolingual sentences with relatively high uncertainty are preferred. Large-scale experiments on WMT English⇒German and English⇒Chinese datasets show that self-training with the proposed uncertainty-based sampling strategy significantly outperforms that with random sampling. Extensive analyses on the generated outputs confirm our claim by concluding that our approach improves the translation of uncertain sentences and the prediction of low-frequency target words.

Our main contributions are:

- We demonstrate the necessity of distinguishing monolingual sentences for self-training.
- We propose an uncertainty-based sampling strategy for self-training, which selects more complementary sentences for the authentic parallel data.
- We show that NMT models benefit more from uncertain monolingual sentences in self-training, which improves the translation quality of uncertain sentences and the prediction accuracy of low-frequency words.

6.2 Preliminary

In this section, we aim to understand the effect of uncertain monolingual data on self-training. We first introduce the metric for identifying uncertain monolingual sentences, then the experimental setup, and at last our preliminary results.

Notations. Let X and Y denote the source and target languages, and let $\mathcal{X}$ and $\mathcal{Y}$ represent the sentence domains of corresponding languages. Let $\mathcal{B} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{N}$ denote the authentic parallel data, where $\mathbf{x}^i \in \mathcal{X}$, $\mathbf{y}^i \in \mathcal{Y}$ and N is the number of sentence pairs. Let $\mathcal{M}_x = \{\mathbf{x}^j\}_{j=1}^{M_x}$ denote the collection of monolingual sentences in the source language, where $\mathbf{x}^j \in \mathcal{X}$ and M_x is the size of the set. Our objective is to obtain a translation model $f : \mathcal{X} \mapsto \mathcal{Y}$, that can translate sentences from language X to language Y.

6.2.1 Identification of Uncertain Data

Data Complexity. According to Zhou et al. [109], the complexity of a parallel corpus can be measured by adding up the translation uncertainty of all source sentences. Formally, the translation uncertainty of a source sentence $\mathbf{x}$ with its translation candidates can be operationalized as conditional entropy:

$$\mathcal{H}(\mathbf{Y}|\mathbf{X} = \mathbf{x}) = -\sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}) \log p(\mathbf{y}|\mathbf{x}) \tag{6.1}$$

$$\approx \sum_{t=1}^{T_x} \mathcal{H}(y|x = x_t), \tag{6.2}$$

where T_x denotes the length of the source sentence, x and y represent a word in the source and target vocabularies, respectively. Generally, a high $\mathcal{H}(\mathbf{Y}|\mathbf{X} = \mathbf{x})$ denotes that a source sentence $\mathbf{x}$ would have more possible translation candidates.

Equation (6.2) estimates the translation uncertainty of a source sentence with all possible translation candidates in the parallel corpus. It can not be directly applied to the sentences in monolingual data due to the lack of corresponding translation candidates. One potential

solution to the problem is utilizing a trained model to generate multiple translation candidates. However, generation may lead to bias estimation due to the generation diversity issue [125, 126]. More importantly, generation is extremely time-consuming for large-scale monolingual data.

Monolingual Uncertainty. To address the problem, we modify Equation (6.2) to reflect the uncertainty of monolingual sentences. We estimate the target word distribution conditioned on each source word based on the authentic parallel corpus, and then use the distribution to measure the translation uncertainty of the monolingual sample. Specifically, we measure the uncertainty of monolingual sentences based on the bilingual dictionary.

For a given monolingual sentence $\mathbf{x}^j \in \mathcal{M}_x$, its uncertainty U is calculated as:

$$\mathrm{U}(\mathbf{x}^j|\mathcal{A}_b) = \frac{1}{T_x}\sum_{t=1}^{T_x}\mathcal{H}(y|\mathcal{A}_b, x = x_t), \tag{6.3}$$

which is normalized by T_x to avoid the length bias. A higher value of U indicates a higher translation uncertainty of the monolingual sentence.

In Equation (6.3), the word level entropy $\mathcal{H}(y|\mathcal{A}_b, x = x_t)$ captures the translation modalities of each source word by using the bilingual dictionary $\mathcal{A}_b$. The bilingual dictionary records all the possible target words for each source word, as well as the respective translation probabilities. It can be built from the word alignments by external alignment toolkits on the authentic parallel corpus. For example, given a source word x with all three word translations y_1, y_2 and y_3 and the translation probabilities of $p(y_1|x)$, $p(y_2|x)$ and $p(y_3|x)$, respectively, the word level entropy can be calculated as follows:

$$\mathcal{H}(y|\mathcal{A}_b, x_i) = -\sum_{y_j \in \mathcal{A}_b(x_i)} p(y_j|x_i)\log p(y_j|x_i). \tag{6.4}$$

6.2.2 Experimental Setup

Data. We conduct experiments on two large-scale benchmark translation datasets, i.e., WMT English⇒German (En⇒De) and

WMT English⇒Chinese (En⇒Zh). The authentic parallel data for the two tasks consists of about 36.8M and 22.1M sentence pairs, respectively. The monolingual data we use is from newscrawl released by WMT2020. We combine the newscrawl data from the year 2011 to 2019 for the English monolingual corpus, consisting of about 200M sentences. We randomly sample 40M monolingual data for En⇒De and 20M for En⇒Zh unless otherwise stated. We adopt newstest2018 as the validation set and used newstest2019 and newstest2020 as the test sets. For En⇒De, we apply BPE [102] with 32K merge operations. As for En⇒Zh, we also applied BPE to English but *Jieba*[1] segmentation to Chinese.

Model. We choose the state-of-the-art TRANSFORMER [15] network as our model, which consists of an encoder of 6 layers and a decoder of 6 layers. We adopt the open-source toolkit Fairseq [103] to implement the model. We use the TRANSFORMER-BASE model for preliminary experiments and the constrained scenario for efficiency. For the unconstrained scenario, we adopt the TRANSFORMER-BIG model. Results on these models with different capacities can also reflect the robustness of our approach. For the TRANSFORMER-BASE model, we train it for 150K steps with 32K (4096×8) tokens per batch. For the TRANSFORMER-BIG model, we train it for 30K steps with 460K (3600×128) tokens per batch with the cosine learning rate schedule [99]. We use 16 Nvidia V100 GPUs to conduct the experiments and select the final model by the best perplexity on the validation set.

Evaluation. We evaluate the models by BLEU score [127] computed by SacreBLEU [128][2]. For the En⇒Zh task, we add the option `-tok zh` to SacreBLEU. We measure the statistical significance of improvement with paired bootstrap re-sampling [104] using `compare-mt`[3] [105].

[1] https://github.com/fxsjy/jieba

[2] `BLEU+case.mixed+lang.[Task]+numrefs.1+smooth.exp++test.wmt[Year]+tok.[Tok]+version.1.4.14`, Task=en-de/en-zh, Year=19/20, Tok=13a/zh

[3] https://github.com/neulab/compare-mt

6.2.3 Effect of Uncertain Data

First of all, we investigate the effect of monolingual data uncertainty on the self-training performance in NMT. We conduct the preliminary experiments on the WMT En⇒De dataset with the TRANSFORMER-BASE model. We sample 8M bilingual sentence pairs from the authentic parallel data and randomly sample 40M monolingual sentences for the self-training. To ensure the quality of synthetic parallel data, we train a TRANSFORMER-BIG model for translating the source monolingual data to the target language. We generate translations using beam search with beam width 5, and follow Edunov et al. [71][4] to remove sentences longer than 250 words as well as sentence-pairs with a source/target length ratio exceeding 1.5. The "teacher" NMT model for self-training is the TRANSFORMER-BIG model to ensure the quality of synthetic data.

Self-training v.s. Data Size. We take a look at the performance of standard self-training and its relationship with data size. Figure 6.1 shows the results. Obviously, self-training with 8M synthetic data can already improve the NMT performance by a significant margin (36.2 averaged BLEU score on WMT En⇒De newstest2019 and newstest2020). It demonstrates the advantage of such a semi-supervised setting over the conventional supervised setting. Further increasing the size of added monolingual data does not bring much more benefit. With all the 40M monolingual sentences, the final performance achieves only 36.5 BLEU points. It indicates that adding more monolingual data only is not a promising way to improve self-training, and more sophisticated approaches for exploiting the monolingual data are desired.

Self-training v.s. Uncertainty. In this experiment, we first adopt *fast-align*[5] to establish word alignments between source and target words in the authentic parallel corpus and use the alignments to build the bilingual dictionary $\mathcal{A}_b$. Then we use the bilingual dictionary

[4] https://github.com/pytorch/fairseq/tree/master/examples/backtranslation
[5] https://github.com/clab/fast_align

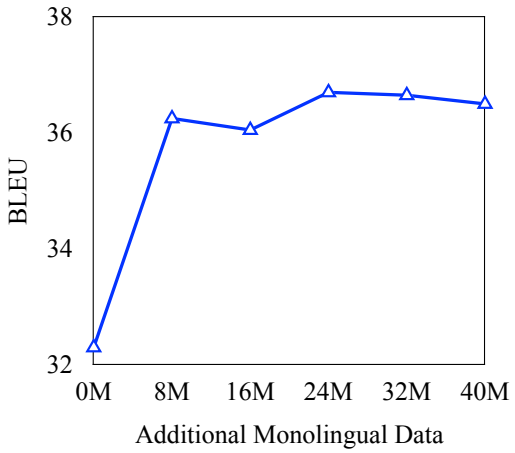


Figure 6.1: Performance of self-training with increased size of monolingual data. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020.

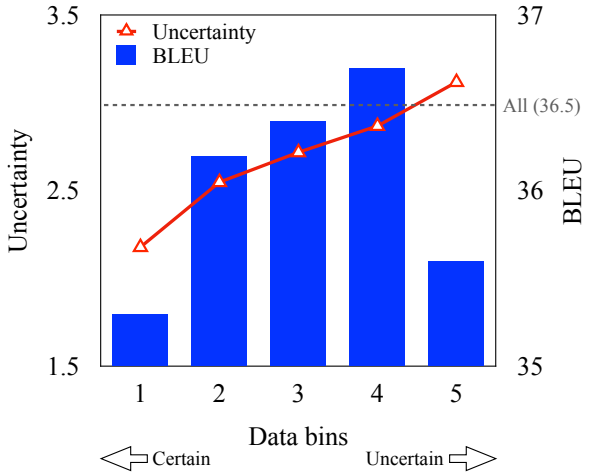
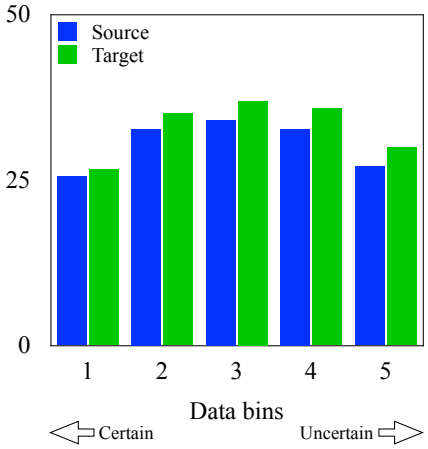


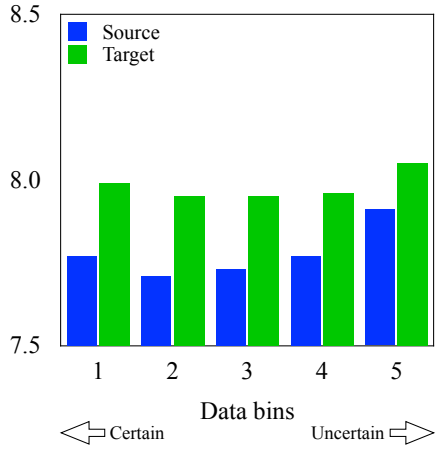
Figure 6.2: Relationship between uncertainty of monolingual data and the corresponding NMT performance. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020.

to compute the data uncertainty expressed in Equation (6.3) for the sentences in the monolingual data set. After that, we rank all the 40M monolingual sentences and group them into 5 equally-sized bins (i.e., 8M sentences per bin) according to their uncertainty scores. At last, we perform self-training with each bin of monolingual data.

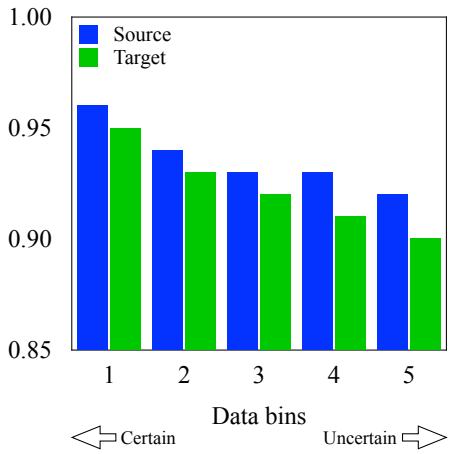
We report the translation performance in Figure 6.2. As seen, there is a trend of performance improvement with the increase of monolingual data uncertainty (e.g., bins 1 to 4) until the last bin. The last bin consists of sentences with excessively high uncertainty, which may contain erroneous synthetic target sentences. Training



(a) Sentence Length



(b) Word Rarity



(c) Coverage

Figure 6.3: Comparison of monolingual sentences with varied uncertainty in terms of three properties, including sentence length, word rarity, and coverage.

on these sentences forces the models to over-fit on these incorrect synthetic data, resulting in the confirmation bias issue [129]. These results corroborate with prior studies [21, 124] such that learning on certain samples brings little gain while on the excessively uncertain samples may also hurt the model training.

6.2.4 Linguistic Properties of Uncertain Data

We further analyze the differences between the monolingual sentences with varied uncertainty to gain a deeper understanding of the uncertain data. Specifically, we perform linguistic analysis on the five data bins in terms of three properties:

- *Sentence Length*: We count the number of tokens in each sentence. Sentence length can reflect the difficulty of sentences as longer sentences may contain more complex expressions.
- *Word Rarity*: We calculate the word rarity [10] to measure the frequency of words in a sentence with a higher value indicating a more rare sentence. The word rarity of a sentence is calculated as follows:

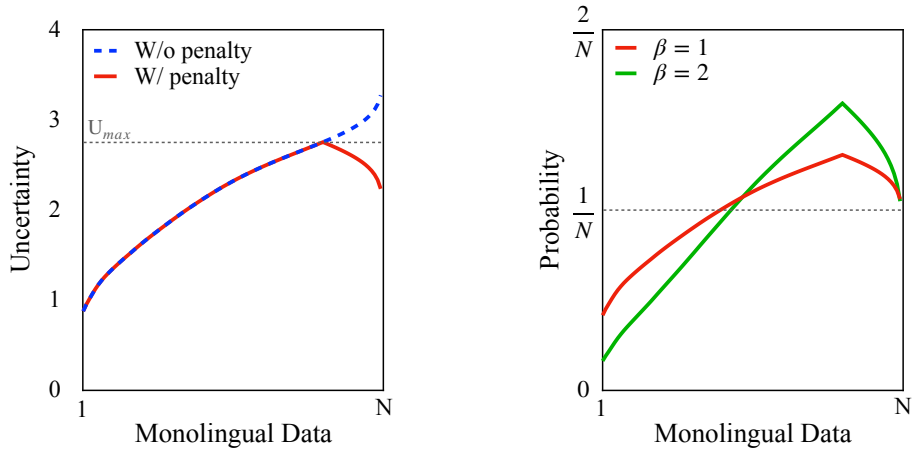
$$\mathrm{WR}(\mathbf{x}) = -\frac{1}{T_x}\sum_{t=1}^{T_x}\log p(x_t), \tag{6.5}$$

  where $p(x_t)$ represents the normalized frequency of word x_t in the authentic parallel data, and T_x is the sentence length.
- *Translation Coverage*: Coverage [130, 107] measures the ratio of source words being aligned with any target words. Firstly, we train an alignment model on the authentic parallel data by *fast-align*[6]. Then we use the alignment model to force-align the monolingual sentences and the synthetic target sentences. Finally, we calculate the coverage of each source sentence.

Among the three properties, the first two reflect the features of monolingual sentences while the last one reflects the quality of synthetic sentence pairs. We calculate these properties for each sentence at the source side and report the averaged results for each data bin in Figure 6.3. We also present the results of the synthetic target sentences for reference.

The results are reported in Figure 6.3. For the length property, we find that monolingual sentences with higher uncertainty are usually longer except for those with excessively high uncertainty (e.g., bin 5). The monolingual sentences in the last data bin noticeably contain more rare words than other bins in Figure 6.3(b), and the rare words in the sentences pose a great challenge in the NMT training process [131]. In Figure 6.3(c), the overall coverage in bin 5 is the lowest among the self-training bins. In contrast, bin 1 with the lowest uncertainty has the highest coverage. These observations suggest that

[6] https://github.com/clab/fast{_}align



(a) Uncertainty (b) Sampling Probability

Figure 6.4: Distribution of modified monolingual uncertainty and sampling probability. The sample with high uncertainty has more chance to be selected while that with excessively high uncertainty would be penalized.

monolingual sentences in bin 1 indeed contain the easiest patterns while monolingual sentences in bin 5 are the most difficult ones, which may explain their relatively weak performance in Figure 6.2.

6.3 Methodology

By analyzing the effect of monolingual data uncertainty on self-training in Section 6.2, we understand that monolingual sentences with relatively high uncertainty are more informative while also with high quality, which motivates us to emphasize the training on these sentences. In this section, we introduce the uncertainty-based sampling strategy for self-training and the overall framework.

6.3.1 Uncertainty-based Sampling Strategy

With the aforementioned measure of monolingual data uncertainty in Section 6.2.1, we propose the uncertainty-based sampling strategy for self-training, which prefers to sample monolingual sentences with relatively high uncertainty.

To ensure the data diversity and avoid the risk of being dominated by the excessively uncertain sentences, we sample monolingual

sentences according to the uncertainty distribution with the highest uncertainty penalized. Specifically, given a budget of N_s sentences to sample, we set two hyper-parameters to control the sampling probability as follows:

$$p = \frac{\left[\alpha \cdot \mathrm{U}(\mathbf{x}^j|\mathcal{A}_b)\right]^\beta}{\sum_{\mathbf{x}^j \in \mathcal{M}_x} \left[\alpha \cdot \mathrm{U}(\mathbf{x}^j|\mathcal{A}_b)\right]^\beta}, \tag{6.6}$$

$$\alpha = \begin{cases} 1, & \mathrm{U}(\mathbf{x}^j|\mathcal{A}_b) \leq \mathrm{U}_{max} \\ max(\frac{2\mathrm{U}_{max}}{\mathrm{U}(\mathbf{x}^j|\mathcal{A}_b)} - 1, 0), & \text{otherwise} \end{cases}, \tag{6.7}$$

where α is used to penalize excessively high uncertainty over a maximum uncertainty threshold U_{max} (Figure 6.4(a)), the power rate β is used to adjust the distribution such that a larger β gives more probability mass to the sentences with high uncertainty (Figure 6.4(b)).

It is difficult to determine whether a sentence is too uncertain. Therefore, we rely on the original parallel data and define a maximum uncertainty threshold U_{max} to penalize the sentence with excessively high uncertainty. Specifically, we sort the source sentences of the original parallel data by their uncertainty in the ascending order and pick the uncertainty of the sentence at the $R\%$ position of all the sentences as U_{max}. R is assumed to be as high as 80 to 100. Because for monolingual data with uncertainty higher than this threshold, they may not be translated correctly by the "teacher" model as there are inadequate such sentences in the authentic parallel data for the model to learn. As a result, monolingual sentences with uncertainty higher than U_{max} should be penalized in terms of the sampling probability.

6.3.2 Overall Framework

Figure 6.5 presents the framework of our uncertainty-based sampling for self-training, which includes four steps:

- Train a "teacher" NMT model and an alignment model on the authentic parallel data simultaneously.

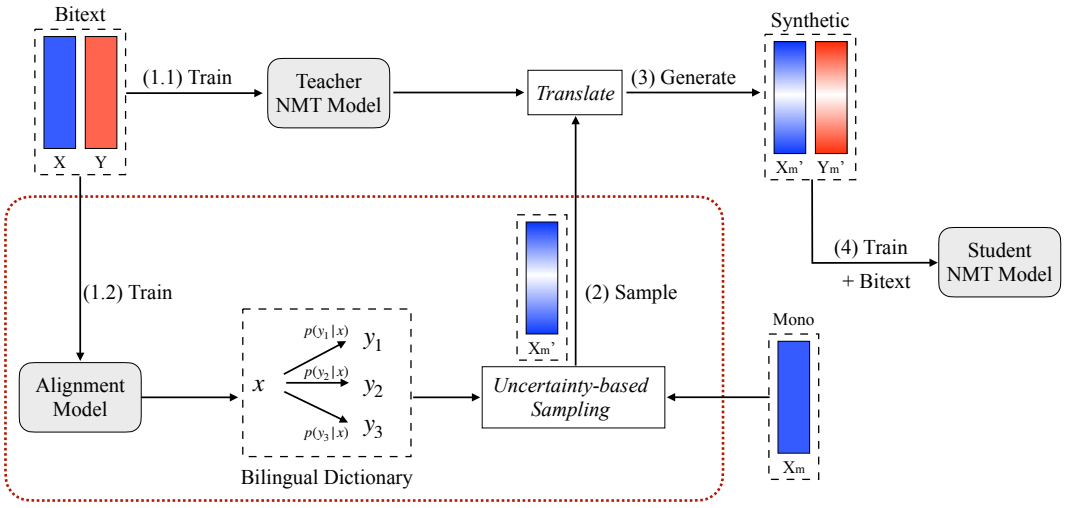


Figure 6.5: Framework of the proposed uncertainty-based sampling strategy for self-training. Procedures framed in the red dashed box corresponds to our approach integrated into the standard self-training framework. "Bitext", "Mono", "Synthetic" denotes authentic parallel data, monolingual data and synthetic parallel data, respectively.

- Extract the bilingual dictionary from the alignment model and perform uncertainty-based sampling for monolingual sentences.
- Use the "teacher" NMT model to translate the sampled monolingual sentences to construct the synthetic parallel data.
- Train a "student" NMT model on the combination of synthetic and authentic parallel data.

6.4 Experiment

6.4.1 Constrained Scenario

We first validate the proposed sampling approach in a constrained scenario, where we follow the experimental configuration in Section 6.2.3 with the Transformer-Base model, the 8M bitext, and the 40M monolingual data. It allows the efficient evaluation of our approach with varied combinations of hyper-parameters and also the comparison with related methods. Specifically, we perform our approach by sampling 8M sentences from the 40M monolingual data and then combining the corresponding 8M synthetic data with the 8M bitext

Table 6.1: Translation performance with respect to different values of β and R. The BLEU score is averaged on WMT En⇒De newstest2019 and newstest2020.

BLEU		R		
		100	90	80
β	1	36.6	36.7	36.6
	2	36.7	**36.9**	36.6
	3	36.5	36.5	36.5

to train the TRANSFORMER-BASE model.

Table 6.1 reports the impact of β and R on the BLEU score. As shown, sampling with high uncertainty sentences and penalizing those with excessively high uncertainty improves translation performance from 36.6 to 36.9. In these experiments, the uncertainty threshold U_{max} for penalizing are 2.90 and 2.74, which are determined by the 90% and 80% (R=90 and 80 in Table 6.1) most certain sentences in the authentic parallel data, respectively. Obviously, the proposed uncertainty-based sampling strategy achieves the best performance with R at 90 and β at 2. In the following experiments, we use $R = 90$ and $\beta = 2$ as the default setting for our sampling strategy if not otherwise stated.

Effect of Sampling. Some researchers may doubt that the final translation quality is affected by the quality of the teacher model. Therefore, translations of high-uncertainty sentences should contain many errors, and it is better to add the results of oracle translations to discuss the sampling effect and the quality of pseudo-sentences separately. To dispel the doubt, we still use the aforementioned 8M bitext as the bilingual data, and use the rest of WMT19 En-De data (28.8M) as the held-out data (with oracle translations) for sampling.

The results are listed in Table 6.2. Clearly, our uncertainty-based sampling strategy (UNCSAMP) outperforms the random sampling strategy (RANDSAMP) when manual translations are used (Rows 2 vs. 3), demonstrating the effectiveness of our sampling strategy based on the uncertainty. Another interesting finding is that using the pseudo-sentences outperforms using the manual translations (Rows 4 vs. 2, 5

Table 6.2: Comparison of our UncSamp and RandSamp with manual translations (Ora: manual translations; ST: pseudo-sentences) on WMT En⇒De newstest2019 and newstest2020.

#	Data	2019	2020	Avg
1	Bitext	36.9	27.7	32.3
2	+ RandSamp Ora	37.4	28.0	32.7
3	+ UncSamp Ora	37.8	28.2	33.0
4	+ RandSamp ST	40.0	30.1	35.0
5	+ UncSamp ST	40.4	30.5	**35.4**

Table 6.3: Comparison of the proposed uncertainty-based sampling strategy with related methods on WMT En⇒De newstest2019 and newstest2020.

Data	2019	2020	Avg
RandSamp	40.9	31.6	36.2
DWF	39.6	30.1	34.8
SrcLM	41.1	32.0	36.5
UncSamp	41.6	32.3	36.9
+ Filtering	41.5	32.7	**37.1**

vs. 3). One possible reason is that the Transformer-big model to construct the pseudo-sentences was trained on the whole WMT19 En-De data that contains the held-out data, which serves as self-training to decently improve the supervised baseline [38].

Comparison with Related Work. We compare our sampling approach with two related works, i.e., difficult word by frequency [DWF, 31] and source language model [SrcLM, 30]. The former one is proposed for monolingual data selection for back-translation, in which sentences with low-frequency words are selected to boost the performance of back-translation. The latter one is proposed for in-domain data selection with in-domain language models. For DWF, we rank the monolingual data by word rarity [10] of sentences and also select the top 80M monolingual data for self-training. For SrcLM, we train an n-gram language model [65][7] on the source sentences in the bitext and measure the distance between each monolingual sentence to the bitext source sentences by cross-entropy. Similarly, we select

[7] https://kheafield.com/code/kenlm/

Table 6.4: Translation performance on WMT En⇒De and WMT En⇒Zh test sets. The results are reported with de-tokenized case-sensitive SacreBLEU. We adopt the TRANSFORMER-BIG with large batch training [11] to achieve the strong performance. "↑ / ⇑": indicate statistically significant improvement over RANDSAMP $p < 0.05/0.01$ respectively.

System	Data	En⇒De			En⇒Zh		
		2019	**2020**	**Avg**	**2019**	**2020**	**Avg**
Wu et al. [76]	BITEXT	37.3	–	–	–	–	–
	+RANDSAMP	39.8	–	–	–	–	–
Shi et al. [132]	BITEXT	–	–	–	–	38.6	–
	+RANDSAMP	–	–	–	–	41.9	–
This Work	BITEXT	39.6	31.0	35.3	37.1	42.5	39.8
	+RANDSAMP	41.6	33.1	37.3	37.6	43.8	40.7
	+SRCLM	41.7	33.1	37.4	37.3	44.0	40.7
	+UNCSAMP	42.5⇑	34.4⇑	**38.4**	38.2⇑	44.3↑	**41.3**

8M monolingual data with the lowest cross-entropy for self-training.

Table 6.3 lists the results. For DWF, it brings no improvement over RANDSAMP, indicating that the technique developed for back-translation may not work for self-training. As for SRCLM, it achieves a marginal improvement over RANDSAMP. The proposed UNCSAMP approach outperforms the baseline RANDSAMP by +0.7 BLEU point, which demonstrates the effectiveness of our approach. In addition to our UNCSAMP approach, we also utilized another N-gram language model at the target side to further filter out the synthetic data with potentially erroneous target sentences. By filtering out 20% sentences from the sampled 8M sentences, our UNCSAMP approach achieves a further improvement up to +0.9 BLEU point.

6.4.2 Unconstrained Scenario

We extend our sampling approach to the unconstrained scenario, where the scale of data and the capacity of NMT models for self-training are increased significantly. We conduct experiments on the high-resource En⇒De and En⇒Zh translation tasks with all the authentic parallel data, including 36.8M sentence pairs for En⇒De

and 22.1M for En⇒Zh, respectively. For monolingual data, we consider all the 200M English newscrawl monolingual data to perform sampling. We train the TRANSFORMER-BIG model for experiments.

Table 6.4 lists the main results of large-scale self-training on high-resource language pairs. As shown, our TRANSFORMER-BIG models trained on the authentic parallel data achieve the performance competitive with or even better than the submissions to WMT competitions. Based on such strong baselines, self-training with RANDSAMP improves the performance by +2.0 and +0.9 BLEU points on En⇒De and En⇒Zh tasks respectively, demonstrating the effectiveness of the large-scale self-training for NMT models. With our uncertainty-based sampling strategy UNCSAMP, self-training achieves further significant improvement by +1.1 and +0.6 BLEU points over the random sampling strategy, which demonstrates the effectiveness of exploiting uncertain monolingual sentences.

6.5 Analysis

In this section, we conduct analyses to understand how the proposed uncertainty-based sampling approach improved the translation performance. Concretely, we analyze the translation outputs of WMT En⇒De newstest2019 from the TRANSFORMER-BIG model in Table 6.4.

6.5.1 Uncertain Sentences

As we propose to enhance high uncertainty sentences in self-training, one remaining question is whether our UNCSAMP approach improves the translation quality of high uncertainty sentences. Specifically, we rank the source sentences in the newstest2019 by the monolingual uncertainty, and divide them into three equally sized groups, namely Low, Medium and High uncertainty.

The translation performance on these three groups is reported in Table 6.5. The first observation is that sentences with high uncertainty

Table 6.5: Translation performance on uncertain sentences. The relative improvements over BITEXT for UNCSAMP are also presented.

Unc	Bitext	RandSamp	UncSamp	
			BLEU	△(%)
Low	38.1	39.7	41.5	8.9
Med	34.2	36.7	37.4	9.3
High	31.0	33.4	34.4	**10.9**

are with relatively low BLEU scores (i.e., 31.0), indicating the higher difficulty for NMT models to correctly decode the source sentences with higher uncertainty. Our UNCSAMP approach improves the translation performance on all sentences, especially on the sentences with high uncertainty (+10.9%), which confirms our motivation of emphasizing the learning on uncertain sentences for self-training.

6.5.2 Low-Frequency Words

Partially motivated by Fadaee et al. [31], we hypothesize that the addition of monolingual data in self-training has the potential to improve the prediction of low-frequency words at the target side for the NMT models. Therefore, we investigate whether our approach has a further boost to the performance on the prediction of low-frequency words. We calculate the word accuracy of the translation outputs with respect to the reference in newstest2019 by `compare-mt`. Following Wang et al. [100], we divide words into three categories based on their frequency, including High: the most 3,000 frequent words; Medium: the most 3,001-12,000 frequent words; Low: the other words.

Table 6.6 lists the results of word accuracy on these three groups evaluated by F-measure. First, we observe that low-frequency words in BITEXT are more difficult to predict than medium and high-frequency words (i.e., 52.3 v.s. 65.2 and 70.3), which is consistent with Fadaee et al. [31]. Second, adding monolingual data by self-training improves the prediction performance of low-frequency words. Our UNCSAMP approach outperforms RANDSAMP significantly on the low-frequency words. These results suggest that emphasizing the learning

Table 6.6: Prediction accuracy of low-frequency words in the translation outputs. The relative improvements over BITEXT for UNCSAMP are also presented.

Freq	Bitext	RandSamp	UncSamp	
			Fmeas	△(%)
Low	52.3	53.8	54.7	**4.5**
Med	65.2	66.5	66.9	2.6
High	70.3	71.6	72.0	2.4

on uncertain monolingual sentences also brings additional benefits for the learning of low-frequency words at the target side.

6.6 Summary

In this chapter, we study the inter-sample quality for uncertainty-based self-training sampling to utilize the large-scale monolingual data more efficiently. We demonstrate the necessity of distinguishing monolingual sentences for self-training in NMT, and propose an uncertainty-based sampling strategy to sample monolingual data. By sampling monolingual data with relatively high uncertainty, our method outperforms random sampling significantly on the large-scale WMT English⇒German and English⇒Chinese datasets. Further analyses demonstrate that our uncertainty-based sampling approach does improve the translation quality of high uncertainty sentences and also benefits the prediction of low-frequency words at the target side. Future work includes the investigation on the confirmation bias issue of self-training and the low-frequency issue in self-training sampling. This study is highly related to our previous study on inactive samples. Both studies utilize the technique of self-training (i.e., forward-translation) and the uncertain sentences could be the inactive samples to some extent. The sentences with excessively high uncertainty should be inactive samples that cannot be easily rejuvenated by forward-translation and more sophisticated approaches need to be developed in the future.

□ **End of chapter.**

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Existing NLP systems are built on three basic components, namely, data, model, and algorithm. Among the three, data is the foundation of NLP systems, since it decides the architecture of models, the scale of models, and the corresponding optimization algorithms. Thus, it has been critical to keep improving the training effectiveness on data for better performance. In this thesis, we present our exploitation of data in two dimensions, i.e., the intra-sample structure and the inter-sample quality. We define intra-sample structure exploitation as fully utilizing the structure information shared by all text samples while inter-sample quality exploitation as emphasizing a certain type of text samples. Regarding intra-sample structure exploitation, we work on the emotion recognition in conversations (ERC) task due to the rich structure information in conversations and focus on context enhancement and self-supervised learning. For inter-sample quality exploitation, we work on neural machine translation (NMT) for the large-scale benchmark datasets and its complete evaluation criteria and focus on uncertainty-based data rejuvenation and data sampling.

In Chapter 3, we exploit the intra-sample structure for context enhancement on the ERC task. It is important to capture the context information accurately to perform the ERC task as an utterance may express different emotions in different contexts. Therefore, we

propose a hierarchical gated recurrent unit (HiGRU) framework with a lower-level GRU to capture word-level contexts and an upper-level GRU to capture utterance-level contexts. We further promote the vanilla HiGRU to two variants, HiGRU with individual features fusion (HiGRU-f) and HiGRU with self-attention and features fusion (HiGRU-sf), so that the word/utterance-level individual inputs and the long-range context information can be sufficiently utilized. Experiments on three widely used ERC datasets demonstrate that our HiGRU models are effective in improving the performance.

In Chapter 4, we exploit the intra-sample structure to perform self-supervised learning on unlabeled conversation data. Data scarcity is an issue for the ERC task, for which annotating more data is costly and time-consuming. To take advantage of the massive unlabeled conversation data, we propose a conversation completion (ConvCom) task to pre-train a context-dependent encoder (Pre-CODE). Essentially, such a pre-training task utilizes the sequential relationship between utterances in each conversation, enabling the learning of representations at both utterance- and conversation-level. Through fine-tuning on labeled data, our Pre-CODE achieves constantly significant improvements across the ERC datasets, and particularly benefits the prediction of minority emotion classes.

In Chapter 5, we exploit the inter-sample quality for uncertainty-based data rejuvenation on the NMT task with large-scale datasets. Large-scale datasets are important for training large and well-performing models, but they also contain complex patterns and potential noises that pose challenges for training NMT models effectively. Thus, we identify the inactive data which contributes less to the model performance by model uncertainty, and show that the existence of inactive data mainly depends on the data distribution. We further introduce data rejuvenation (DataReju) to improve the training of NMT models on large-scale datasets by relabeling the inactive data with forward-translation. Experiments on large-scale datasets show that our DataReju approach consistently and significantly improves performance with strong NMT models.

In Chapter 6, we exploit the inter-sample quality for uncertainty-based self-training sampling to leverage monolingual data more efficiently for the NMT task. Self-training augments the training of NMT models with synthetic parallel data, which is usually constructed from a randomly sampled subset of large-scale monolingual data. However, we empirically show random sampling is sub-optimal and propose to select the most informative monolingual sentences for self-training. Specifically, we calculate the translation uncertainty of monolingual sentences based on a bilingual dictionary from the authentic parallel data. Then we propose an uncertainty-based sampling (UncSamp) strategy that prefers to sample monolingual sentences with relatively higher uncertainty for self-training. Experiments on large-scale NMT datasets demonstrate that our UncSamp improves the translation quality, especially for uncertain sentences, and the prediction accuracy of low-frequency words at the target side.

In summary, this thesis studies the effective training of NLP models with data engineering in the intra-sample structure and inter-sample quality perspectives. Extensive experiments on benchmark datasets confirm the effectiveness and efficiency of our proposed approaches.

7.2 Future Work

In the future, we are particularly interested in how external monolingual data improves the translation quality of low-frequency words. Generally, monolingual data can be leveraged in the form of semi-supervised learning (e.g., data augmentation, multitask learning) or self-supervised learning (e.g., pre-training and fine-tuning). We elaborate the low-frequency words issue in terms of these directions.

7.2.1 Low-Frequency Issue in Data Augmentation

Due to the less occurrence in the training data, low-frequency words cannot be learned well by NMT models. As a result, the "Teacher" models, while could be strong, are still weak at translating low-

frequency words. Besides, sequence generation by the current beam search algorithm tends to favor high-frequency words, which will worsen the training of low-frequency words in data augmentation techniques like back-translation [70] and self-training [38]. According to our studies in Chapter 5 and Chapter 6, we also try to emphasize the learning of medium-to-low frequency words, because the high frequency-words have already been learned well. Particularly in Chapter 6, we empirically show that selecting low-frequency words for self-training cannot make improvements to the translation quality. Therefore, how to improve the quality of synthetic data containing low-frequency words is critical to further improve existing data augmentation methods. A possible direction is to incorporate the statistic machine translation [96] or bilingual lexicon induction [133], which proves more effective in translating low-frequency words.

7.2.2 Low-Frequency Issue in Multilingual Machine Translation

The ultimate goal of machine translation is to develop a unified multilingual machine translation system that can translate any language. It is also common knowledge that training multiple language pairs jointly can significantly improve the translation quality of low-resource language pairs [134, 135]. However, high-resource language pairs usually encounter a noticeable performance drop relative to that in bilingual training. We are curious about the low-frequency issue for two reasons. First, while the low-frequency issue has been extensively discussed in bilingual training, it has not been well investigated in multilingual training, which may explain the different behaviors of improvements on low- and high-resource language pairs. Second, multilingual training adopts a large vocabulary for multiple languages, which may further deteriorate the low-frequency issue. Third, recent studies show that universal representations learning [136, 137] and multitask learning [138] are beneficial to multilingual training, which we believe the improvements should be related to low-frequency words. Clearly, it is important to study these research questions in

order to better understand and improve existing multilingual machine translation systems.

7.2.3 Self-Supervised Multilingual Pre-training

We believe self-supervised multilingual pre-training is a promising way to alleviate the low-frequency issue. Self-supervised learning performed by the pre-training and fine-tuning paradigm has now become a standard configuration in natural lanaguge process (NLP), e.g., ELMo [24], GPT [26] and BERT [8] for natural language understanding (NLU), and MASS [139], BART [140], and mBART [37] for natural language generation (NLG). Different from data augmentation, pre-training does not introduce erroneous synthetic data, which prevents error propagation when training student models. Instead, pre-training exploits the self-supervised signals in monolingual data to learn representations that can be inherited by downstream tasks to improve performance. Pre-training has proven particularly effective in low-resource translations [37], which exactly alleviates the low-frequency issue for multilingual machine translation.

However, current pre-training approaches like mBART still only learn representations of sentences in each language although the data from different languages are used for training jointly. Recent studies construct code-switched data by the substitution of word-level translation lexicons in order to inject alignment signals into monolingual data (e.g., CSP [141]). For phrase-level alignment signals, an n-gram bilingual phrase table has been utilized for cross-lingual pre-training (i.e., CMLM [142]) to benefit unsupervised NMT (UNMT). Further, Wang et al. [143] tries to utilize high-resource bilingual data as the sentence-level alignment signal to improve the performance of UNMT. In the area of multilingual pre-training for NMT, it is still desired to learn better representations for the cross-attention module in the sequence-to-sequence framework by enhancing phrase- and sentence-level alignment signals.

□ **End of chapter.**

Bibliography

[1] Yoon Kim, "Convolutional neural networks for sentence classification," in *EMNLP*, 2014.

[2] Christopher Olah, "Understanding LSTM networks," *Colah's Blog*, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu, "Neural machine translation: A review of methods, resources, and tools," *AI Open*, 2020.

[4] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy, "Hierarchical attention networks for document classification," in *NAACL-HLT*, 2016.

[5] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency, "Context-dependent sentiment analysis in user-generated videos," in *ACL*, 2017.

[6] Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann, "Conversational memory network for emotion recognition in dyadic dialogue videos," in *NAACL-HLT*, 2018.

[7] Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander F. Gelbukh, and Erik Cambria, "Dialoguernn: An attentive RNN for emotion detection in conversations," in *AAAI*, 2019.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[9] Devamanyu Hazarika, Soujanya Poria, Roger Zimmermann, and Rada Mihalcea, "Conversational transfer learning for emotion recognition," *Information Fusion*, vol. 65, 2021.

[10] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell, "Competence-based curriculum learning for neural machine translation," in *NAACL*, 2019.

[11] Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato, "Analyzing uncertainty in neural machine translation," in *ICML*, 2018.

[12] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *EMNLP*, 2017.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," *NeurIPS*, 2014.

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin, "Convolutional sequence to sequence learning," in *ICML*, 2017.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[16] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli, "Scaling neural machine translation," in *WMT*, 2018.

[17] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao, "Learning deep transformer models for machine translation," in *ACL*, 2019.

[18] Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao, "Norm-based curriculum learning for neural machine translation," in *ACL*, 2020.

[19] M. Pawan Kumar, Benjamin Packer, and Daphne Koller, "Self-paced learning for latent variable models," in *NeurIPS*, 2010.

[20] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.

[21] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *NeurIPS*, 2017.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *NeurIPS*, 2013.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[24] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018.

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

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, "Improving language understanding by generative pre-training," 2018.

[27] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language models are unsupervised multitask learners," 2019.

[28] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell *et al.*, "Language models are few-shot learners," 2020.

[29] Philipp Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT Summit*, 2005.

[30] Robert C. Moore William Lewis, "Intelligent selection of language model training data," *ACL*, 2010.

[31] Marzieh Fadaee and Christof Monz, "Back-translation sampling by targeting difficult words in neural machine translation," in *EMNLP*, 2018.

[32] Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun, "Improving back-translation with uncertainty-based confidence estimation," in *EMNLP*, 2019.

[33] Tom Kocmi and Ondřej Bojar, "Curriculum learning and minibatch bucketing in neural machine translation," in *RANLP*, 2017.

[34] Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat, "An empirical exploration of curriculum learning for neural machine translation," *arXiv*, 2018.

[35] Wei Wang, Isaac Caswell, and Ciprian Chelba, "Dynamically composing domain-data selection with clean-data selection by "co-curricular learning" for neural machine translation," in *ACL*, 2019.

[36] Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao, "Uncertainty-aware curriculum learning for neural machine translation," in *ACL*, 2020.

[37] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer,

"Multilingual denoising pre-training for neural machine translation," *TACL*, vol. 8, 2020.

[38] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato, "Revisiting self-training for neural sequence generation," in *ICLR*, 2019.

[39] Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R. Lyu, "HiGRU: Hierarchical gated recurrent units for utterance-level emotion recognition," in *NAACL-HLT*, 2019.

[40] Wenxiang Jiao, Michael R. Lyu, and Irwin King, "Exploiting unsupervised data for emotion recognition in conversations," in *EMNLP: Findings*, 2020.

[41] Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael R. Lyu, and Zhaopeng Tu, "Data rejuvenation: Exploiting inactive training examples for neural machine translation," in *EMNLP*, 2020.

[42] Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael R. Lyu, and Irwin King, "Self-training sampling with monolingual data uncertainty for neural machine translation," in *ACL*, 2021.

[43] Abien Fred Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv*, 2018.

[44] Peng Wang, Jiaming Xu, Bo Xu, Chenglin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao, "Semantic clustering and convolutional neural network for short text categorization," in *ACL-IJCNLP*, 2015.

[45] Ye Zhang and Byron C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," in *IJCNLP*, 2017.

[46] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[47] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[48] Duyu Tang, Bing Qin, and Ting Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *EMNLP*, 2015.

[49] Florian Eyben, Martin Wöllmer, and Björn W. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *ACMMULTIMEDIA*, 2010.

[50] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," in *ICML*, 2010.

[51] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus, "End-to-end memory networks," in *NeurIPS*, 2015.

[52] Wenxiang Jiao, Michael R. Lyu, and Irwin King, "Real-time emotion recognition via attention gated hierarchical memory network," in *AAAI*, 2020.

[53] Jingye Li, Donghong Ji, Fei Li, Meishan Zhang, and Yijiang Liu, "Hitrans: A transformer-based context-and speaker-sensitive model for emotion detection in conversations," in *COLING*, 2020.

[54] Haiqin Yang and Jianping Shen, "Emotion dynamics modeling via bert," in *IJCNN*, 2021.

[55] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang, "Hitransformer: Hierarchical interactive transformer for efficient and effective long document modeling," in *ACL*, 2021.

[56] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, 2003.

[57] Tomas Mikolov, Jiri Kopecky, Lukas Burget, Ondrej Glembek *et al.*, "Neural network based language models for highly inflective languages," in *ICASSP*, 2009.

[58] Holger Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, 2007.

[59] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv*, 2013.

[60] Frederic Morin and Yoshua Bengio, "Hierarchical probabilistic neural network language model." in *AISTATS*, vol. 5, 2005.

[61] Andriy Mnih and Koray Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *NeurIPS*, 2013.

[62] Ryo Masumura, Naoki Makishima, Mana Ihori, Akihiko Takashima, Tomohiro Tanaka, and Shota Orihashi, "Large-context conversational representation learning: Self-supervised learning for conversational documents," in *IEEE SLT*, 2021.

[63] Haozhe Liu, Hongzhan Lin, and Guang Chen, "Tantp: Conversational emotion recognition using tree-based attention networks with transformer pre-training," in *PAKDD*, 2021.

[64] Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba, "Denoising neural machine translation training with trusted data and online data selection," in *WMT*, 2018.

[65] Kenneth Heafield, "KenLM: Faster and smaller language model queries," in *EMNLP*, 2011.

[66] Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio, "Semantic redundancies in image-classification datasets: The 10% you don't need," *arXiv*, 2019.

[67] Alex Krizhevsky, Geoffrey Hinton *et al.*, "Learning multiple layers of features from tiny images," *Tech Report*, 2009.

[68] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[69] Runzhe Zhan, Xuebo Liu, Derek F Wong, and Lidia S Chao, "Meta-curriculum learning for domain adaptation in neural machine translation," in *AAAI*, 2021.

[70] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Improving neural machine translation models with monolingual data," in *ACL*, 2016.

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

[72] Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli, "On the evaluation of machine translation systems trained with back-translation," in *ACL*, 2020.

[73] Isaac Caswell, Ciprian Chelba, and David Grangier, "Tagged back-translation," in *WMT*, 2019.

[74] Benjamin Marie, Raphael Rubino, and Atsushi Fujita, "Tagged back-translation revisited: Why does it really work?" in *ACL*, 2020.

[75] Jiajun Zhang and Chengqing Zong, "Exploiting source-side monolingual data in neural machine translation," in *EMNLP*, 2016.

[76] Lijun Wu, Yiren Wang, Yingce Xia, QIN Tao, Jianhuang Lai, and Tie-Yan Liu, "Exploiting monolingual data at scale for neural machine translation," in *EMNLP*, 2019.

[77] Xuan-Phi Nguyen, Shafiq Joty, Wu Kui, and Ai Ti Aw, "Data diversification: An elegant strategy for neural machine translation," *arXiv*, 2019.

[78] Chao-Chun Hsu and Lun-Wei Ku, "Socialnlp 2018 emotionx challenge overview: Recognizing emotions in dialogues," in *SocialNLP@ACL*, 2018.

[79] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth Narayanan, "IEMOCAP: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, 2008.

[80] Sayyed M. Zahiri and Jinho D. Choi, "Emotion detection on TV show transcripts with sequence-based convolutional neural networks," in *AAAI*, 2018.

[81] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *ACL*, 2019.

[82] Amir Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *ACL*, 2018.

[83] Pierre Lison and Jörg Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *LREC*, 2016.

[84] Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu, "Multi-turn response selection for chatbots with deep attention matching network," in *ACL*, 2018.

[85] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[86] Paul Ekman, *Universal and cultural differences in facial expressions of emotion.* Lincoln: University of Nebraska Press, 1971.

[87] Paul Ekman, "Are there basic emotions?" *Psychological Review*, vol. 99, no. 3, 1992.

[88] Sopan Khosla, "EmotionX-AR: CNN-DCNN autoencoder based emotion classifier," in *SocialNLP@ACL*, 2018.

[89] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[90] Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou, "Reinforced mnemonic reader for machine reading comprehension," in *IJCAI*, 2018.

[91] Linkai Luo, Haiqing Yang, and Francis Y. L. Chin, "EmotionX-DLC: Self-attentive bilstm for detecting sequential emotions in dialogues," in *SocialNLP@ACL*, 2018.

[92] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[93] Edmund Tong, Amir Zadeh, Cara Jones, and Louis-Philippe Morency, "Combating human trafficking with multimodal deep models," in *ACL*, 2017.

[94] Michael Gutmann and Aapo Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.

[95] Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao K. Huang, and Lun-Wei Ku, "Emotionlines: An emotion corpus of multi-party conversations," in *LREC*, 2018.

[96] Philipp Koehn and Rebecca Knowles, "Six challenges for neural machine translation," in *WMT*, 2017.

[97] Aviral Kumar and Sunita Sarawagi, "Calibration of encoder decoder models for neural machine translation," *arXiv*, 2019.

[98] Tobias Domhan, "How much attention do you need? a granular analysis of neural machine translation architectures," in *ACL*, 2018.

[99] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli, "Pay less attention with lightweight and dynamic convolutions," in *ICLR*, 2019.

[100] Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu, "On the Inference Calibration of Neural Machine Translation," in *ACL*, 2020.

[101] Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou, "Confidence-aware scheduled sampling for neural machine translation," in *ACL*, 2021.

[102] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016.

[103] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "Fairseq: A fast, extensible toolkit for sequence modeling," in *NAACL (Demonstrations)*, 2019.

[104] Philipp Koehn, "Statistical Significance Tests for Machine Translation Evaluation," in *EMNLP*, 2004.

[105] Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang, "compare-mt: A tool for holistic comparison of language generation systems," in *NAACL (Demonstrations)*, 2019.

[106] Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi *et al.*, "Findings of the 2019 conference on machine translation," in *WMT*, 2019.

[107] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li, "Modeling coverage for neural machine translation," in *ACL*, 2016.

[108] Chris Dyer, Victor Chahuneau, and Noah A. Smith, "A simple, fast, and effective reparameterization of ibm model 2," in *NAACL*, 2013.

[109] Chunting Zhou, Graham Neubig, and Jiatao Gu, "Understanding knowledge distillation in non-autoregressive machine translation," in *ICLR*, 2019.

[110] Liang Ding, Longyue Wang, Xuebo Liu, Derek F Wong, Dacheng Tao, and Zhaopeng Tu, "Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation," in *ACL*, 2021.

[111] Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *NeurIPS*, 2017.

[112] Jinlong Liu, Guoqing Jiang, Yunzhi Bai, Ting Chen, and Huayan Wang, "Understanding why neural networks generalize well through gsnr of parameters," in *ICLR*, 2020.

[113] Guanlin Li, Lemao Liu, Guoping Huang, Conghui Zhu, and Tiejun Zhao, "Understanding data augmentation in neural machine translation: Two perspectives towards generalization," in *EMNLP-IJCNLP*, 2019.

[114] Shuo Wang, Zhaopeng Tu, Zhixing Tan, Shuming Shi, Maosong Sun, and Yang Liu, "On the language coverage bias for neural machine translation," in *ACL (Findings)*, 2021.

[115] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu, "Semi-supervised learning for neural machine translation," in *ACL*, 2016.

[116] Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen, "Joint training for neural machine translation models with monolingual data," in *AAAI*, 2018.

[117] Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu, "Neural machine translation with monolingual translation memory," in *ACL*, 2021.

[118] Aditya Siddhant, Linting Xue, Melvin Johnson, Mihir Sanjay Kale, Noah Constant, and Rami Al-Rfou, "nmt5-is parallel data still relevant for pre-training massively multilingual language models?" in *ACL*, 2021.

[119] Shuo Ren, Long Zhou, Shujie Liu, Furu Wei, Ming Zhou, and Shuai Ma, "Semface: Pre-training encoder and decoder with a semantic interface for neural machine translation," in *ACL*, 2021.

[120] Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li *et al.*, "Achieving human parity on automatic chinese to english news translation," *arXiv*, 2018.

[121] Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang, "Baidu neural machine translation systems for wmt19," in *WMT*, 2019.

[122] Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov, "Facebook fair's wmt19 news translation task submission," in *WMT*, 2019.

[123] Yoon Kim and Alexander M. Rush, "Sequence-level knowledge distillation," in *EMNLP*, 2016.

[124] Subhabrata Mukherjee and Ahmed Awadallah, "Uncertainty-aware self-training for few-shot text classification," in *NeurIPS*, 2020.

[125] Jiwei Li, Will Monroe, and Dan Jurafsky, "A simple, fast diverse decoding algorithm for neural generation," *arXiv*, 2016.

[126] Raphael Shu, Hideki Nakayama, and Kyunghyun Cho, "Generating diverse translations with sentence codes," in *ACL*, 2019.

[127] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL*, 2002.

[128] Matt Post, "A call for clarity in reporting BLEU scores," *WMT*, 2018.

[129] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *IJCNN*, 2020.

[130] Shahram Khadivi and Hermann Ney, "Automatic filtering of bilingual corpora for statistical machine translation," in *NLDB*, 2005.

[131] Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu, "Token-level adaptive training for neural machine translation," in *EMNLP*, 2020.

[132] Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao, "Oppo's machine translation systems for WMT20," in *WMT*, 2020.

[133] Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime G Carbonell, "Domain adaptation of neural machine translation by lexicon induction," in *ACL*, 2019.

[134] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv*, 2016.

[135] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas,

Martin Wattenberg, Greg Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *TACL*, 2017.

[136] Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li, "Pre-training multilingual neural machine translation by leveraging alignment information," in *EMNLP*, 2020.

[137] Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li, "Contrastive learning for many-to-many multilingual neural machine translation," in *ACL*, 2021.

[138] Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Xu Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu, "Leveraging monolingual data with self-supervision for multilingual neural machine translation," in *ACL*, 2020.

[139] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu, "MASS: Masked sequence to sequence pre-training for language generation," in *ICML*, 2019.

[140] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020.

[141] Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju, "CSP: Code-switching pre-training for neural machine translation," in *EMNLP*, 2020.

[142] Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma, "Explicit cross-lingual pre-training for unsupervised machine translation," in *EMNLP-IJCNLP*, 2019.

[143] Mingxuan Wang, Hongxiao Bai, Hai Zhao, and Lei Li, "Cross-lingual supervision improves unsupervised neural machine translation," *arXiv*, 2020.