Extending Generative NLP: Incorporating Diversity, Context, and Inclusivity in Neural Text Generation

Kaushal Kumar Maurya

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Doctor of Philosophy

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Computer Science and Engineering

March 2, 2024

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

Kaushal Kumar Maurya

(Name)

CS18RESCH11003

(Roll No.)

Approval Sheet

This thesis titled – "Extending Generative NLP: Incorporating Diversity, Context, and Inclusivity in Neural Text Generation" by (Mr. Kaushal Kumar Maurya) is approved for the degree of Doctor of Philosophy from IIT Hyderabad.

Prof. Ganesh Ramakrishnan
Department of CSE,
IIT Bombay
Examiner 1

Prof. Monojit Choudhury
Department of ~~CSE,~~ NLP
Mohamed bin Zayed University of Artificial Intelligence
Examiner 2

V.B. Vineeth
Prof. Vineeth Balasubramanian
Department of CSE,
IIT Hyderabad
Internal Examiner

Maunendra
Dr. Maunendra Sankar Desarkar,
Department of CSE,
IIT Hyderabad
Adviser/Guide

Prof. J. Balasubramaniam
Department of Mathematics,
IIT Hyderabad
Chairman

Acknowledgements

Back in 2018, when I started the PhD program, I underestimated the magnitude of this rollercoaster journey. Undoubtedly, this has been filled with many ups and downs like others. This thesis is the outcome of the last 5.5 years of effort with the help and support of my advisors, collaborators, friends, and family.

I could not have had such a wonderful experience in my PhD journey without the wisest and kindest advisor, Dr. Maunendra Sankar Desarkar. When I first met with him, I was struggling to adjust to the IIT system, worrying about courses/grades, and having a little experience in research - so the fact that Dr. Maunendra gave me a chance without asking any investigative questions (not making me more nervous) was amazing. I got to know more about Dr. Maunendra in the last 5.5 years; he is calm, composed, humble, and strives to do the right thing and to do it well. In this way, he has inspired me and given me the freedom to become a better scientist, teacher, community member, and leader. He is not only a wonderful advisor but also a wonderful person who teaches you to balance life, value yourself and others, and make decisions wisely. I always feel more relaxed after each meeting with Dr. Maunendra; thank you for your easy availability. Especially during the COVID-19 time and the last two years of my Ph.D., there were many unavoidable family commitments and urgent requests. I am so grateful to Dr. Maunendra for caring about students' well-being and being supportive. This has been such a less stressful and quality research-focused journey. Thank you so much, Dr. Maunendra, for having the best influence on this period of my life and my career.

Thank you to my doctoral review committee - Profs. Vineeth N. Balasubramanian, Dr. Srijith P.K., and Prof. J. Balasubramaniam - whose continual feedback and encouragement have been invaluable throughout my time at IIT Hyderabad. They are all role models to me, and I will follow their suggestions/directions for the rest of my life.

I have been fortunate to work with many wonderful research collaborators at Microsoft, Nvidia, Shizuoka University, and the University of Tokyo. Bharat Kumar, Deepshikha Kumari, Prof. Yoshinobu Kano, Prof. Akiko Aizawa, Dr. Manish Gupta, Puneet Agarwal, Dr. Anoop Kunchukuttan, Amit, Diksha, Rahul, Rohit, and the entire MT team at Hyderabad - these colleagues and mentors have helped me better understand how to do research. A special thank you to Anoop for being more than a collaborator, a wonderful mentor, and a friend.

I extend my gratitude to my wonderful collaborators and friends from IIT Hyderabad - Dr. Sreekanth Madisetty, Arkadipta De, Venkatesh Elangovan, Aditi Bagora, Kamal Shrestha, Deepak Kumar, Maharaj Brahma, and Aishwarya M. It has been enjoyable working especially with Dr. Sreekanth, Venkatesh, Maharaj, and Aishwarya. I have learned a lot with them by working closely, spending a lot of time on research discussions, late-night brainstorming sessions, facing failures and successes together, and learning from each other.

One of the most rewarding things I've done at IIT Hyderabad is being part of NLP courses as a teaching assignment. Thank you, Dr. Maunendra, for the opportunities to take lectures, hands-on sessions, and project mentoring.

Next, I want to thank the each of Natural Language Information Processing lab members with whom I have witnessed the transformative period of NLP Research. Samujjwal Ghosh, Suvodip Dey, Shounak Kundu, Manisha Dubey, Debolena Basak, Kaushiki Dwivedi, Shrey Satapara, Sharan Narasimhan, Vikram Anand Singh and Priyambada Ambastha — who helped and inspired me to work and explore new things.

I express my gratitude to Microsoft and Nvidia for providing opportunities for research internships. Microsoft has played a crucial role as a key collaborator throughout my PhD journey. This collaboration took various forms, including support through internships, collaboration with different teams (such as auto-suggest and machine translation), mentors for research discussions, acceptance of our MAPG grant for two consecutive years, sharing computational resources, providing access to confidential data for research purposes, and support for two in-person conference travels (ACL 2023 and ECML-PKDD 2023). I am greatly thankful to Microsoft.

I am grateful to the *CL conferences and ACL community for making all conference resources publicly accessible, supporting many conference participantion by waiving registration fees, and providing travel support through the ACL diversity and inclusion program. This has been extremely helpful in maximizing my engagement, learning, collaboration, discussions, and attending conferences.

My research theme is generative NLP, which lacks reliable automated evaluation metrics and mostly relies on human evaluation. I extend my deepest gratitude to all the human annotators - the list is quite extensive - for their time to work on my urgent requests. Gratitude to all the anonymous reviewers and meta-reviewers for your valuable feedback to improve the papers. Special thanks to Venkat, Maharaj, Dr. Anoop, Aishwarya, Tejasri, Dr. Suhash, Dr. Prasen, and Dr. Maunendra for reading many early drafts of write-ups and providing their valuable feedback, which helped improve the drafts.

I express my gratitude to the former heads of the CSE department, Prof. M. V. Panduranga Rao and Dr. Subrahmanyam Kalyanasundaram, as well as the current head, Prof. Antony Franklin of CSE, for placing their trust in me, providing an opportunity to pursue PhD and offering research facilities. I would like to extend a special acknowledgment to Prof. M. V. Panduranga Rao for his love, kindness, care, and support in adjusting in the initial stages of my PhD. I also appreciate the CSE office and the entire staff for being my initial point of contact and offering assistance with various requirements.

I am grateful to the former director, Prof. U. B. Desai, and current Director, Prof. B. S. Murthy, for fostering a research-focused environment at IIT Hyderabad.

I am indebted to Mr. Jessy Bondla, Mr. Imtiaz Ahmed, Mr. Vijay Chakravarthi, and the computer center team for their continuous support with networking and server issues.

Going back even further, I also want to thank my school teacher, Shivanand Singh, and undergraduate faculty, Rajesh Pandey, for all the guidance, love, support, and encouragement they provided during my pursuit of the PhD. Furthermore, I express my gratitude to my postgraduate (MTech) thesis advisor, Dr. K.N. Murthy, for solidifying my research foundation.

I have made so many fantastic friends over the course of my PhD. Deepest gratitude to Dr. Utpal Bora, Dr. Sailaja Rahanala, Ritesh Gupta, Rohit Aggarwal, Madhukar Misra, Palkesh Sanklecha, Suraj, Priyambada Ambastha, Vikram Anand Singh, Ankit, Kaushiki Dwivedi, Tejasri N, Rambabu D, Dr. Sakshi Varshney, Subbu, Appu, Shashank, Harish S A, K Shiv Kumar, Vishnuprasadh Kumaravelu, Rishabh Karnad, Siddharth Jain, Nilesh Shah, S. VenkataKeerthy, and many more for their love, quality time, and support throughout my PhD.

Lastly, I would like to express my gratitude to my family. Firstly, to my wife, Sweta Singh Maurya: she has been a constant source of inspiration, emotional support, and a partner in decision-making discussions. She has shared similar phases of ups and downs, always cheering me on. I am truly grateful to have her in my life. Most importantly, I thank my parents and grandparents, who provided me with a stress-free environment, freedom, and financial support throughout this long journey. From the first day of school to the last day of the PhD, they have consistently been positive and encouraging. My siblings, Rani, Soni, and Kishan, have always been my number one fans, eager to learn and follow. I also extend my sincere thanks to my wife's family: Neha, Amit, Nidhi, Divya, Deepak, and Mother-in-law for consistently making me feel valued and encouraging me to continue pursuing my passion.

Dedication

To my grandparents, my parents, my siblings, my wife, and my wife's family.

दादा-दादी, माई-बाबूजी, भाई-बहिन, पत्नी अवुरी पत्नी के परिवार के।

Abstract

Advancements in deep learning have yielded remarkable success in Natural Language Generation (NLG), driven by advancements in neural architectures and the availability of large datasets. However, the wide adoption of these NLG models for downstream tasks is often challenging, especially in scenarios such as *applications requiring diverse text generation*, *limited context in data*, and *limited volume of task-specific labeled data.* Diverse text generation necessitates a *one-to-many* setup, where the model generates multiple outputs that are semantically similar yet lexically diverse, all derived from a single input. In the limited context scenario, the model often generates unexpected output due to the lack of relevant context in the input text. The limited data scenario is a frequent and more challenging problem, particularly for low-resource languages (LRLs). Current NLP research has primarily focused on high-resource languages (HRLs), e.g., English, which benefit from computationally accessible large training data. Despite the exciting progress in HRLs, there are over 7,000 languages globally, and the majority lack the necessary resources to train modern deep neural networks. In fact, collecting labeled data for these LRLs is often prohibitively expensive or infeasible. The scarcity of task-specific labeled data is more pronounced for NLG tasks, which limits the extension of NLG technology to LRLs.

In this thesis, we address the aforementioned challenges and extend NLG modeling to diverse text generation, limited context, and limited data (i.e., low-resource languages) scenarios. This thesis contains two parts. The first part addresses the *diverse text generation and limited context issues.* In particular, we have designed a semantic decoupling and multi-decoder-based approach to guide diverse text generation. Further, we explore the retrieval-augmented generation (RAG) type of modeling approach to augment relevant external context in deep neural networks to address limited context issues. The second part of the thesis is dedicated to *extending NLG modeling to LRLs.* Here, we focus on cross-lingual modeling - transferring supervision from HRLs to LRLs. Our primary focus is on zero-shot modeling for scalability. In particular, we first focus on well-formed zero-shot text generation in LRLs by mitigating the catastrophic forgetting problem. We achieve this through unsupervised adaptive training. Next, we propose a novel meta-learning-based approach to transfer more uniform cross-lingual supervision across multiple LRLs and NLG tasks. Finally, we extend NLG modeling for extremely low-resource languages (ELRLs) that lack parallel data, have no or limited monolingual data, and are absent in modern large multilingual pre-trained language models. To achieve this, we propose noise augmentation techniques inspired by surface-level lexical similarity between *closely-related*

HRLs and ELRLs. These proposed modeling approaches successfully overcome the mentioned limitations and extend NLG modeling to benefit a wider population.

Contents

List of Tables xv

List of Figures xx

List of Algorithms xxiii

1 Introduction **1**

1.1 Challenges and Thesis Objectives 3

1.2 Main Contributions 6

1.3 Thesis Outline 8

2 Background **11**

2.1 Introduction to Natural Language Generation 11

2.1.1 Defining Natural Language Generation Task 11

2.1.2 Relevant NLG Tasks 12

2.1.3 Evaluation Metrics 15

2.2 Neural Networks for NLG Tasks 17

2.2.1 RNN Sequence-to-Sequence Neural Network 19

2.2.2 Transformers Sequence-to-Sequence Neural Network 20

2.2.3 Tokenization Techniques 20

2.2.4 Training and Evaluation of NLG Neural Network 22

2.3 Pre-training of Neural Network Models 22

2.3.1 Language Models 23

2.3.2 Pre-training and Fine-Tuning of Language Models 24

2.3.3 Multilingual Pre-training of Language Models 25

2.3.4 Relevant Pre-trained Language Models 26

3 Advancing Frontiers of NLG: Distractor Generation **27**

3.1 Introduction 27

3.2 Related Work 29

3.2.1 Traditional Approaches 30

3.2.2 Learning Based Approaches 30
3.3 Problem Statement 31
3.4 Methodology 31
3.4.1 Model Overview 31
3.4.2 Hierarchical Encoder 32
3.4.3 Hierarchical Multi-Decoder 38
3.4.4 Training and Dis-Similarity Loss 41
3.5 Experimental Setting 42
3.5.1 Dataset 42
3.5.2 Baselines 43
3.5.3 Evaluation Metrics 44
3.5.4 Implementation Details 45
3.6 Results And Analysis 45
3.6.1 Automatic Evaluation Results 45
3.6.2 Human Evaluation Results 47
3.6.3 Ablation Study and Inter-distractor Similarity Test 49
3.6.4 Case Study 50
3.7 Conclusion 51
3.8 Insights, Limitations and Future Work 51

4 Advancing Frontiers of NLG: Personalized Query Auto-Completions 53
4.1 Introduction 53
4.2 Related Work 56
4.2.1 Traditional Approaches for QAC 56
4.2.2 Learning-Based Approaches for QAC 56
4.2.3 NLG-Based Approaches for QAC 57
4.3 Problem Formulation 57
4.4 Methodology 58
4.4.1 Trie Context Extraction ($\mathrm{MPC}_{\mathrm{Main}}$) 58
4.4.2 Synthetic Context Extraction ($\mathrm{MPC}_{\mathrm{Synth}}$) 59
4.4.3 Context Augmentations in NLG 60
4.5 Datasets and Experimental Setup 61
4.5.1 Datasets and Analysis 61
4.5.2 Evaluation Metrics 63
4.5.3 Baselines 64
4.5.4 Implementation Details 66

4.6 Results and Discussions . . . 66
4.6.1 Overall Performance Comparison . . . 66
4.6.2 Performance Analysis for Short Prefixes . . . 70
4.6.3 Ablation Study . . . 70
4.6.4 Trie Completion Retention Analysis . . . 71
4.6.5 Runtime Analysis . . . 72
4.6.6 Case Studies . . . 73
4.7 Conclusion . . . 73
4.8 Insights, Limitations and Future Work . . . 74

5 Mitigating Catastrophic Forgetting to Enable Zero-Shot Cross-Lingual Generation 76
5.1 Introduction . . . 76
5.2 Related Work . . . 79
5.3 Methodology . . . 80
5.3.1 Background: Multilingual BART (mBART) . . . 80
5.3.2 Unsupervised Auxiliary Task . . . 82
5.3.3 Freezing Model Components . . . 83
5.3.4 Adding Language Tag . . . 83
5.3.5 Model Training and Generation . . . 84
5.4 Experimental Setup . . . 84
5.4.1 Baselines . . . 84
5.4.2 Evaluation Metrics . . . 85
5.4.3 Implementation Details . . . 86
5.5 Downstream NLG Tasks and Results . . . 87
5.5.1 News Headline Generation (NHG) . . . 87
5.5.2 Question Generation (QG) . . . 89
5.5.3 Abstractive Text Summarization (ATS) . . . 90
5.5.4 Distractor Generation (DG) . . . 90
5.5.5 Overall Results . . . 91
5.6 Further Analyses and Discussions . . . 93
5.6.1 ZmBART Performance for HRL . . . 93
5.6.2 Effect of Auxiliary Task for LRLs . . . 94
5.6.3 Ablation Study for Catastrophic Forgetting Problem . . . 95
5.6.4 Effect of Architecture on Few-shot Training . . . 96
5.6.5 Few-shot Performance vs. Supervised Data Size . . . 96

5.7 Conclusion 97
5.8 Insights, Limitations and Future Work 97

6 Meta-Learning Approach to Improve Zero-Shot Cross-Lingual Transfer and Generation 99

6.1 Introduction 99
6.2 Related Work 101
6.2.1 Cross-Lingual Generation 101
6.2.2 Meta-Learning for NLP 102
6.3 Background: Meta-Learning (MAML) 102
6.4 Methodology 103
6.4.1 Language Clustering 104
6.4.2 Meta-X_{NLG} Training 105
6.4.3 Avoiding Catastrophic-Forgetting (CF)/Accidental Translation (AT) Problem: 107
6.5 Experiment Setup 108
6.5.1 Tasks and Datasets 108
6.5.2 Question Generation (QG): 109
6.5.3 Baselines 110
6.5.4 Evaluation Metrics 111
6.5.5 Implementation Details 113
6.6 Results and Analyses 113
6.6.1 Cross-lingual Transfer: 114
6.6.2 Effect of Different Centroid (Meta-Train) Languages: 115
6.6.3 Case Study 116
6.7 Conclusion 118
6.8 Insights, Limitations and Future Work 118

7 Zero-Shot Machine Translation for Extremely Low-resource Languages 121

7.1 Introduction 121
7.2 Related Work 125
7.2.1 MT for Low-resource Languages/Dialects 125
7.2.2 Vocabulary Adaptation for Low-resource MT 126
7.2.3 Data Augmentation for Low-resource MT 126
7.3 Character Span Noise Augmentation 127
7.3.1 Methodology 127

7.3.2 Experimental Setup 128
7.3.3 Results and Analyses 132
7.3.4 Further Analyses and Discussions 137
7.3.5 Summary 140
7.4 Selective Character Noise Augmentation 140
7.4.1 Methodology 140
7.4.2 Experimental Setup 146
7.4.3 Results and Discussions 150
7.4.4 Summary 155
7.5 Conclusion 156
7.6 Insights, Limitations and Future Work 157

8 Conclusion and Future Directions 158
8.1 Summary of Contributions 158
8.2 Key Ideas and Suggestions 160
8.3 The Value of Research 161
8.4 Future Directions 161

References 164

List of Tables

2.1 Sample triplet and corresponding distractors from Race dataset [LXL+17] for DG task. 12

2.2 Sample session, prefix, and top-5 completions from Bing search engine for QAC task. '||' is search query separator in session. 13

2.3 Sample document and corresponding summary from XL-Sum dataset [HBI+21] for ATS task 13

2.4 Sample news article and corresponding headline from PENS dataset [AWL+21] for NHG task. 13

2.5 Sample passage, answer and question from SQuAD dataset [RZLL16a] for QG task 14

3.1 Statistics of RACE and RACE++ DG dataset. Average statistics are computed across all three splits. 43

3.2 Automatic evaluation results on the RACE-DG dataset. For Seq2Seq and HCA, the results are taken from [GBL+19]. For HRED and HSA (rows with *), the results are taken from [ZLW20] as these numbers are better than the numbers reported in the original paper [GBL+19] for the same dataset. Symbol (-) indicates that results are not available. 46

3.3 Automatic evaluation results of different models on RACE++ DG dataset. 46

3.4 Comparative study results of human evaluation. 47

3.5 Quantitative study results of human evaluations 49

3.6 Ablation study results of the first distractor on the RACE DG dataset. Where EEL is the evidence Encoding Layer, CR is contextual representation, CER is contextual evidence representation (gated mechanism output), DSL dis-similarity loss, and QCI question Context Initialization. The row with * indicates that -the results are obtained without CER where the SS matching score is obtained from the average of CR and EEL scores 50

3.7 Average Jaccard Similarity scores across generated three distractors on RACE-DG dataset. 50

4.1 Prefix distribution statistics for Bing dataset with prefix character length. ‘M’ and ‘K’ indicate that the value is in the order of millions and thousands respectively. 62

4.2 Prefix distribution statistics for AOL dataset with prefix character length. ‘M’ and ‘K’ indicate that the value is in the order of millions and thousands respectively. 62

4.3 Results of the models on Bing dataset. The reported scores are percentage (%) improvements over $\text{MPC}_{\text{Train}}$ + $\text{MPC}_{\text{Synth}}$ baseline. ‘-’ indicates no completions are retrieved/generated for the model. Here we consider up to 3 completions as additional context from MPC_{Main} or $\text{MPC}_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built. 67

4.4 Results of the models on AOL dataset. The reported scores are percentage (%) improvements over $\text{MPC}_{\text{Train}}$ + $\text{MPC}_{\text{Synth}}$ baseline. ‘-’ indicates no completions are retrieved/generated for the model. Here we consider up to 3 completions as additional context from MPC_{Main} or $\text{MPC}_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built. 67

4.5 Results of the models on AOL dataset. Here we report exact evaluation scores, unlike the ones for the Bing dataset. We consider up to 3 completions as additional context from MPC_{Main} or $\text{MPC}_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built. . . . 68

4.6 Performance analysis for short prefixes. For Bing, % improvements over $\text{MPC}_{\text{Train}}$+$\text{MPC}_{\text{Synth}}$ are reported. For AOL, actual scores are reported. 70

4.7 Results of ablation study using different experimental setups. TRIE-NLG(m) means “TRIE-NLG + Up to Top-m Completions”. For the Bing dataset, percentage improvements over $\text{MPC}_{\text{Train}}$+$\text{MPC}_{\text{Synth}}$ baseline are reported. For the AOL dataset, actual evaluation scores are reported. 71

4.8 Number of examples where t trie suggestions were retained in the TRIE-NLG generated completions for *Seen Test Datasets*. 72
4.9 Percentage of times the candidate suggestion from trie was copied to [1-8]th positions (‘Pos’) as output by TRIE-NLG for Bing Seen Test Dataset. ‘None’ indicates the candidate suggestion was not a part of TRIE-NLG output. Results are shown for Seen Test Data when m=3. ‘Rank’ indicates the rank of candidate suggestion from trie. 72
4.10 Percentage of times the candidate suggestion from trie was copied to [1-8]th positions (‘Pos’) as output by TRIE-NLG for AOL Seen Test Dataset. ‘None’ indicates the candidate suggestion was not a part of TRIE-NLG output. Results are shown for Seen Test Data when m=3. ‘Rank’ indicates the rank of candidate suggestion from trie. 73
4.11 Runtime of different test dataset splits. Values are in milliseconds(ms)/record for 8 auto-complete generations. 73

5.1 Data preparation steps for the Auxiliary Task 83
5.2 Zero-shot and few-shot results for Hindi language 88
5.3 Zero-shot and few-shot results for Japanese language 88
5.4 Human evaluation results for zero-shot generated outputs in the Hindi language 88
5.5 Human evaluation results for zero-shot generated outputs in the Japanese language 89
5.6 ZmBART model performance for high-resource English language in two setups: without and with the proposed auxiliary task. Results are reported across all four tasks with five automated evaluation metrics. 94
5.7 Zero-shot performance of ZmBART with and without auxiliary task for Hindi and Japanese languages 95
5.8 Different approaches to mitigate the catastrophic forgetting problem for QG task. hi: Hindi, ja: Japanese 95
5.9 Different approaches to mitigate the catastrophic forgetting problem for NHG task. hi: Hindi, ja: Japanese 95
5.10 Few-shot results with different architectural setups for the Hindi language. WD: word embeddings and all decoder parameters are frozen 96

6.1 Clustering of considered 30 Languages 105

6.2 Details of the datasets used in Meta-X$_{\text{NLG}}$. For adaptive pre-training, a small 5k/1k/1k dataset is used. Ada.PT: Adaptive unsupervised pre-training. *-English is a high-resource language for which all three splits were used, as shown in Row 1. **-Additional language added to the dataset. ***-The dataset does not have a validation split, so a test data set of centroid languages is used in training, and the training set is used for evaluation (test set). 111

6.3 Details of language clustering for each dataset, mean cosine distance (meanCD), and centroid languages. For each dataset, we group languages into three clusters as shown in Figure 6.1. The Swahili language does not have any typological or task-based representations, so we added it to cluster 1 based on language typological features and heuristics. For the TyDiQA dataset, only two clusters are obtained as cluster-2 does not have any language. If a cluster has only two languages, we randomly selected any language as a centroid language. 112

6.4 Zero-shot ROUGE-L scores for 19 target LRL on XL-Sum dataset [HBI+21]. EnZmT5 [MDKD21a] and FTZmT5 are baseline models. Scores are reported after an extensive hyperparameter search for all the models. 114

6.5 Zero-shot ROUGE-L scores for 14 target LRLs on Wikilingua dataset [LDCM20a]. 115

6.6 Zero-shot BLEU scores for 8 target LRLs on XQuAD dataset [ARY20a]. . 115

6.7 Zero-shot BLEU scores for 7 target LRLs on TyDiQA data [CCC+20a]. . 115

6.8 Zero-shot BLEU scores for 4 target LRLs on MLQA data [LOR+20a]. . . 116

6.9 Human Evaluation results for four languages (**hi:** Hindi, **te:** Telugu, **ta:** Tamil and **bn:** Bengali), two annotator sets, two tasks (**ATS** and **QG**) and all five datasets. **Flu:** Fluency, **Rel:** Relatedness and **Corr:** Correctness metrics. Results are shown for two annotation sets, which ensure bias-free evaluation. Reported scores are the average of all the annotators in an annotator set. 116

6.10 Meta-X$_{\text{NLG}}$'s zero-shot evaluation scores (Rouge-L) with different meta-training (centroid) language combinations on the XL-Sum dataset. We cut the hierarchical clustering dendrogram shown in Figure 6.2, at the lower level to obtain more clusters. In total, we obtained eight centroid languages, i.e., pa, es, vi, tr, ja, mr, fr and ru. '-' indicates the language used in training, so scores are not zero-shot and not included. Markers '*', '**', and '@' indicate meta training with all-centroid, all-non-centroid, and a mix of both (centroid & non-centroid) languages. 117

7.1 Dataset details and Statistics. * are obtained from [RDB+22] 130
7.2 Model implementation and training details 133
7.3 Zero-shot chrF scores results for ELRLs → English MT. 133
7.4 Zero-shot BLEU scores results for ELRLs → English MT 133
7.5 Zero-shot BLEURT (computed with *BLEURT-20* checkpoint) scores results for ELRLs → English MT 134
7.6 Zero-shot COMET (computed with *Unbabel/wmt22-comet-da* model) scores results for ELRLs → English MT 134
7.7 Zero-shot chrF scores with additional lexically less similar languages. `HRL:` hi and mr; `sim:` lexical similarity 135
7.8 Average cosine similarity between representations of source HRLs and source ELRLs. UNC: Unigram Char-Noise 136
7.9 Analysis experiment to show zero-shot chrF scores with lexically diverse HRLs. Due to computational constraints, we have considered 1 million parallel data for each HRL. 136
7.10 Translation quality (chrF) with an additional 1000 ELRL-English parallel sentences (ELRL_{par}). 137
7.11 BLEU and chrF Scores: HRLs performance for all three language families 138
7.12 BLEURT and COMET Scores: HRLs performance for all three language families 138
7.13 Ablation Study and Different Experimental Setups. Similar trends were observed for other ELRLs and language families. 139
7.14 Grammatical Well-Formedness for different models with L'AMBRE . 140
7.15 Statistics of the language and data used in SelectNoise model . . . 147
7.16 Zero-shot chrF (↑) scores results for ELRLs → English 150
7.17 Zero-shot BLEU (↑) scores results for ELRLs → English 150
7.18 Zero-shot BLEURT (↑) scores results for ELRLs → English 150
7.19 Zero-shot COMET (↑) scores results for ELRLs → English 151
7.20 Human evaluation scores with XSTS metrics 151
7.21 SelectNoise Model performance with larger monolingual data . . . 154
7.22 Zero-shot translation performance of Vanilla NMT vs. SelectNoise on less related LRLs with HRL (Hindi) 155
7.23 Comparative performance for HRLs across both Indo-Aryan and Romance families. 155

List of Figures

1.1 Examples of machine translation (left) and text generation in LRL (right) 3
1.2 Outline of the Thesis 9

2.1 Illustration of zero-shot and few-shot process for low-resource language generation. Multilingual representation and Cross-lingual transfer help in LRL generation. Here, we consider abstractive text summarization as an example NLG task. 14

3.1 Architectural diagram of the proposed Hierarchical Multi-Decoder Model (*better viewed in color*) 32
3.2 Flow diagram of Hierarchical Encoder (*better viewed in color*) 33
3.3 Flow diagram of softsel operation 35
3.4 An overview of distractor generation from HMD-Net model. 51

4.1 Examples of session, prefix and completion. The session can have multiple previously user-typed queries. Here, we consider the session which has heterogeneous queries. Session queries are separated by '‖'. 54
4.2 An overview of the proposed Trie-NLG model 59
4.3 Sample generations from Trie-NLG model. Here we consider two examples: a seen short prefix and an unseen prefix. 74

5.1 Sample zero-shot news headline generation with ZmBART in the Hindi language 78
5.2 Overview of ZmBART developed on top of mBART [LGG+20c] model 81
5.3 Trends of few-shot performance of ZmBART with supervised Hindi and Japanese data for ATS (left) and QG (right) tasks. 96

6.1 An overview of Meta-X$_{\text{NLG}}$ framework 104

6.2 Language clustering is based on a multi-view representation [OHB20]. We intentionally showcase more than 30 languages in the clustering, which will prove useful for scaling the proposed work in the future. The dotted red line represents the cut-line used to obtain different sets of clusters. Here, we show three clusters that perform best in our case. 105

6.3 Cosine distance between language tags obtained from EnZmT5 baseline (left) and Meta-X_{NLG} (right) for 10 langues from XL-Sum dataset. Dark colors indicate a higher cosine distance. 117

6.4 Zero-shot samples generated by Meta-X_{NLG} in Telugu, Tamil, Bengali and Hindi languages. The top three samples are for ATS, and the bottom three are for QG tasks. The generated samples are taken from all five datasets. In some instances, the model learns to generate an actual target language script even though the reference is in transliterated form. See the underlined token (in red font color) in the TyDiQA-Tamil example. 120

7.1 Hindi (HIN; HRL), Bhojpuri (BHO; ELRL) and Chhattisgarhi (HNE; ELRL) parallel sentences. Additionally, the corresponding noisy Hindi example with character-span (CSN) noise. BHO and HNE are closely related to HIN. After noise augmentation, noisy Hindi becomes lexically more similar to BHO and HNE. 123

7.2 Illustration of character noise augmentation with random baseline [AS22] and propose SelectNoise model. The SelectNoise enhances lexical similarity between noisy HRL (N-HIN) and ELRL (BHO). ENG: *English*, HIN: *Hindi*, N-HIN: *Noisy Hindi* and BHO: *Bhojpuri* languages. Red: *Insertion*, Blue: *Deletion* and Green: *Substitution* operations. 124

7.3 Overview of proposed CharSpan model 127

7.4 Candidate alphabets for noise augmentation from ELRLs 128

7.5 Heatmaps showing lexical similarity (LCSR) are presented for three language families. The Indo-Aryan languages have the Devanagari script, whereas languages from the Romance and Malay-Polynesian families have the Latin script. 130

7.6 Lexical similarity heatmap for additional languages used in the analysis section. Here we have shown similarity scores for Assamese (asm), Bengali (ben), Gujrati (guj), Panjabi (pan), Hindi (him), Marathi (mar), Oriya (ory), Malayalam (mal), Kannada (kan), Tamil (tam) and Telugu (tel) languages. 136

7.7 The generation errors (transliteration) from different baseline models. The proposed CharSpan model successfully mitigates those errors. Colors indicate the corresponding transliteration in a generation. . . . 139

7.8 Overview of the proposed selective noise augmentation-based model for extremely low-resource MT. 140

7.9 Illustration of selective candidates extraction for noise augmentation that utilizes BPE merge and edit operations. Here *I*, *D*, and *S* indicate insertion, deletion, and substitution respectively. Frequencies are associated with operations. 0 indicates the corresponding edit operation was not extracted. 143

7.10 *Top* is a template for the character candidate pool $\mathbf{S}_c$. The operations are: *insertion* (I), *deletion* (D) and *substitution* (S). The character c_i represents the i^{th} element of $\mathcal{S}_c$, which is an HRL character, $c'_1 \dots c'_k$ denote the corresponding substituting candidates from ELRLs and f is the associated frequencies. The *Bottom* shows a few sample elements of the $\mathbf{S}_c$. 144

7.11 Lexical similarity heatmap between HRL and its related ELRLs. Fig. (a) depicts a similarity score for the Indo-Aryan family where HRL is Hindi. Fig. (b) depicts a similarity score for the Romance family where HRL is Spanish. *Note: Darker color denotes high lexical similarity.* . 148

7.12 Performance trends of the proposed model with various k and p values from top-k and top-p sampling, respectively. 153

7.13 Language similarity vs. Performance. 154

7.14 Sample translations from various models for ELRLs to English MT direction. RCN: Random Character Noise injection model, UCN: Unsupervised Character Noise injection model (i.e., SelectNoise model) and SCN: Supervised Character Noise injection model. 156

List of Algorithms

1 Meta-X$_{\text{NLG}}$: Meta-Learning Algorithm for Cross-lingual Generation . 107

2 CharSpan: Character-span Noise Augmentation Algorithm 129

3 SelectNoise: Unsupervised Noise Augmentation 142

4 Supervised Noise Augmentation 146

Chapter 1

Introduction

In 1978, Douglas Adams presented his comedy science fiction series *The Hitchhiker's Guide to the Galaxy* through a late-night BBC Radio show. The series portrays the protagonist, Arthur Dent, a human from Earth, who gains the extraordinary ability to comprehensively understand and communicate with various aliens, facilitated by the use of a fictional device called the *Babel fish* inserted into his ear. Now, 45 years after The Hitchhiker's Guide, it is still science fiction to have a real *Babel fish* that enables language technology to approximately 7000 languages that are present across the globe[1]. In recent times, remarkable progress has been made in the field of natural language processing (NLP). Despite this progress, the application of language technologies remains limited to only a few hundred languages, with a particular focus on a few high-resource languages [Ben19, JSB+20].

Extending this discussion, the primary driving force for such remarkable advancement in NLP research has been propelled by large pre-trained language models (PLMs; [ZZL+23, YJT+23]) based on self-supervised training objectives [YJT+23]. These PLMs are developed on top of transformer neural network [VSP+17] and have millions or billions of parameters. They undergo training on large monolingual data for thousands of compute hours, yielding high-quality *out-of-box* representations. Subsequently, these models can be fine-tuned for downstream tasks, leading to superior accuracy. However, there is a notable disparity in NLP research, with the majority of studies being conducted on English data [Ben19, JSB+20], despite the fact that the vast majority of the global population, approximately 95%, do not speak English as their primary language, and a staggering 75% does not speak English at all[2]. Further, there are around 7,000 spoken languages, with approximately

[1] https://www.ethnologue.com/insights/how-many-languages/

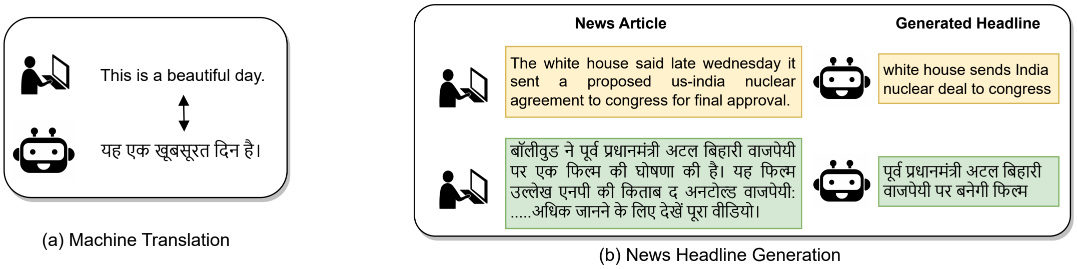
[2] https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

400 languages having over 1 million speakers and about 1,200 languages having more than 100,000 speakers [vELR+22]. As per Joshi et al. [JSB+20], 88% of the world's languages, spoken by 1.2 billion people, and are untouched by the benefits of language technology. A study presented at ACL 2008 [Ben11] revealed that 63% of all papers focused only on English. A more recent study during ACL 2021 [RVS22] concluded that nearly 70% of the papers were evaluated on English datasets. 10 years of progress and language coverage in NLP research is almost unchanged due to the limited availability of datasets for low-resource languages (LRLs), aka. the long tail of languages. To put it succinctly, the scarcity of data, lack of linguistic tools and resources, and absence of representation from PLMs [DRK+21] leads to performance gaps or hinders advancement of language technology for LRLs[3].

This thesis is a step towards enabling language technologies for tailored low-resource languages (LRLs) characterized by limited available data. The primary focus is on natural language generation (NLG), a field concerned with *the automated generation of human-like text from a given input context.* The context can be a natural language text, an image, a video, etc. NLG consists of a wider range of tasks, including machine translation, abstractive text summarization, headline generation, question generation, and many more. The issue of data scarcity is more pronounced for NLG tasks as task-specific data availability for LRLs is even rare. Current multilingual language models support only around 100 languages [XCR+21]. Moreover, their adaptability to various generative applications, even for 100 languages, poses a significant challenge [AHO+23]. The thesis objective is to replicate the capabilities of the *Babel fish* — automated machine translation — for LRLs. Moreover, the scope of this thesis extends beyond machine translation — extending NLG technology to three frequently observed yet challenging scenarios: (1) diverse text generation, (2) text generation with limited context, and (3) text generation with limited labeled data, as is the case for LRLs. Fig. 1.1 presents examples of Hindi-to-English translation and news headline generation in an HRL (i.e., English) and LRL (i.e., Hindi).

Diverse text generation in NLG is crucial for real-world applications. For example, diverse headlines for input news articles can give the author/publisher few options to select and broaden the audience's engagement [EMKD23]. Similarly, generating diverse responses for frequently occurring monotonous contexts in a dialogue system enhances user engagement. Another application is the distractor generation (DG), which involves generating multiple distractors or incorrect options for a given Multiple-Choice Question (MCQ) for reading comprehension, i.e., a triplet of `<pas-`

[3] https://labs.theguardian.com/digital-language-divide/?ref=ruder.io


This is a beautiful day.
यह एक खूबसूरत दिन है।
News Article
Generated Headline
The white house said late wednesday it sent a proposed us-india nuclear agreement to congress for final approval.
white house sends India nuclear deal to congress
बॉलीवुड ने पूर्व प्रधानमंत्री अटल बिहारी वाजपेयी पर एक फिल्म की घोषणा की है। यह फिल्म उल्लेख एनपी की किताब द अनटोल्ड वाजपेयी: .....अधिक जानने के लिए देखें पूरा वीडियो।
पूर्व प्रधानमंत्री अटल बिहारी वाजपेयी पर बनेगी फिल्म

(a) Machine Translation

(b) News Headline Generation

Figure 1.1: Examples of machine translation (left) and text generation in LRL (right)

`sage, question, correct answer>`. This capability is particularly valuable for saving time for course instructors or Intelligent Tutoring Systems (ITS; [Wen14]) for creating diverse incorrect options. The task of distractor generation is one of the problems addressed in the thesis. Taking this forward, any NLG model requires an input context for performing the generation. Sometimes, the input context is limited or non-relevant, posing a challenge to the NLG modeling and leading to meaningless or ambiguous generations. In this thesis, we have considered a *limited context* scenario, frequently observed in personalized query auto-completion (PQAC) tasks, specifically for short and unseen prefixes. It is a task of generating completions given the user-specific `<query prefix, sessions>`. The session comprises previously typed queries within a specified time span. For short and unseen prefixes, the context within the prefix and in the session is limited (non-available or non-relevant), leading to poor-quality completions.

These problems hinder the advancement of NLG modeling. We delineate more fine-grained challenges associated with these problems and establish the thesis objectives based on them.

1.1 Challenges and Thesis Objectives

Context Selection and Diverse Text Generation: *Encoder-decoder-based* NLG models have been explored in literature for various generative applications. We focus on the *distractor generation* task, which involves generating distractors or incorrect options for given Multiple-Choice Questions (MCQ) for reading comprehension, i.e., the input is a triplet of `<passage, question, correct answer>`. The ideal distractors should possess the following properties: (i) contextual relevance to the question, (ii) semantic dissimilarity to the answer, (iii) diversity from each other, and (iv) confusion-inducing. It has been observed that existing models may fail to achieve one or more of these properties. For instance, generated distractors may be similar to the answer [ZLW20] or lack diversity [GBL+19], among other issues. *In this thesis, we propose a modeling approach based on semantic decoupling of passage sentences and a multi-decoder network to generate diverse distractors.*

Generation with Limited Context: In another NLG application, personalized query auto-completions (PQAC), the challenges related to *limited context* are prevalent and result in low-quality generations. Query formulation can be time-consuming for naive or users with complex information needs. Modern search engines, therefore, have a Query Auto-Completion (QAC) module to assist users in efficiently expressing their information needs as a search query. Due to advancements in NLG, QAC is formulated as an NLG problem, which involves generating top-m completions given the user-specific `<query prefix, session>`. The session contains personalized data - previously typed queries - making them Personalized QAC (PQAC) systems.

While research in query completion spans over many decades, the challenge of *limited context*, particularly for short and unseen prefixes, persists. Short prefixes typically consist of a few characters and unseen prefixes are those which have never been recorded previously (new query prefixes typed by the user). The traditional Trie-based model [MC15] offers the most popular completions (MPC) for short prefixes and provides no completions for unseen prefixes. Modern NLG-based models can be used to overcome the limitations by generating completions for unseen prefixes. However, since short prefixes have few characters and unseen prefixes are rarely typed, their context within the prefix and in the session is limited, leading to poor and non-relevant completions. *In this thesis, we address the challenge of generating high-quality completions with limited context, particularly for short and unseen prefixes.*

Catastrophic forgetting Problem in Zero-shot Generation: The remarkable progress in NLP is primarily driven by large annotated datasets. However, most low-resource languages (LRLs) lack such annotated datasets. To address this issue, cross-lingual transfer has emerged as a popular technique for enabling language technology in LRLs with limited supervision or *limited annotated data*. In this modeling approach, a multilingual pre-trained language model (mPLM) is trained with large task-specific data in high-resource languages (HRL) and then evaluated for the task on unseen LRLs (zero-shot) or on LRLs with limited examples (few-shot). This modeling regime transfers supervision from HRLs to LRLs, extending language technology in many LRLs. For example, let us consider a sentiment analysis task. Initially, a mPLM is trained using a large HRL dataset, often in English, where the input consists of English sentences, and the target is the class label. Subsequently, when a sentence in LRL (e.g., in Hindi) is fed, the model generates an appropriate class label for the input sentence.

This modeling recipe presents additional challenges for the NLG tasks. One challenge is the issue of *catastrophic forgetting*. In the zero-shot evaluation phase of the NLG task in LRLs, the text must be generated in LRLs. For instance, the abstractive text summarization model (developed with the above recipe) is expected to generate a zero-shot summary in LRL when given an LRL input article. However, it is observed that [XCR+21] the zero-shot generations are in HRL or code-mixed with HRL and LRL. This occurs because the model forgets the multilingual pre-training, which is referred to as the catastrophic forgetting/off-target/accidental translation problem. This issue has impeded the extension of existing NLG techniques to a wide range of LRLs. *In this thesis, we address the catastrophic forgetting problem to enable the well-formed zero-shot generation in LRLs.*

Non-uniform Cross-lingual Transfer in Zero-shot Generation: The second challenge in cross-lingual modeling is *non-uniform supervision transfer*. As supervision is transferred from HRL to LRLs, for LRLs that are similar (close) to the considered HRLs, the transfer strength is high. However, for languages that are less similar to HRL, the transfer is often weak. This creates issues of non-uniform supervision transfer, which directly impacts the zero-shot performance for LRLs; that is, the better the supervision transfer, the better the performance, and vice versa. *This thesis investigates modeling approaches to transfer supervision from HRL to LRLs more uniformly.*

Extending Generative NLP for Extremely Low-resource Languages: There are approximately 7,000 spoken languages worldwide, ranging from HRLs like English to LRLs such as Hindi or Japanese. Within the spectrum of LRLs, there exists a large subset known as Extremely Low-Resource Languages (ELRLs), exemplified by languages like Bhojpuri or Sundanese. Unlike HRLs and some LRLs, ELRLs lack parallel or pseudo-parallel data, have limited monolingual resources, and are not represented in multilingual pre-trained language models. This scarcity of learning resources makes the task of developing NLG applications for ELRLs more challenging.

This thesis investigates modeling approaches to develop an NLG technology for ELRLs, particularly machine translation from ELRLs to English.

1.2 Main Contributions

Considering the above objectives, we make the following contributions with this thesis:

1. We address the problem of diverse text generation for the distractor generation (DG) task. Specifically, we propose a novel Hierarchical Encoder-Decoder, LSTM-based neural network (HMD-Net; [MD20b]) model for the distractor generation. On the encoder side, it employs *softsel* and *Gated Mechanism* to identify candidate (decouple) sentences in the input passage that are not semantically similar to the answer and maintain context with the question. The decoder conditions on these candidate sentences, the question statement, and the correct answer in a multi-decoder setup (interconnected with each other) to generate lexically diverse and confusing multiple distractors. We further utilize linguistic features and BERT contextual token embedding representations to boost the model's performance. We prepared a new DG dataset from the existing RACE MCQ dataset. The proposed model achieved, on average, 10.99% BLEU and a huge 70.27% ROUGE-L improvement across three distractors over the best baseline.

2. We address the limited context problem in personalized query auto-completions (PQAC), specifically for *short* and *unseen* prefixes. We leverage insights from both Trie and NLG and proposed TRIE-NLG model [MDGA23]. In TRIE-NLG, we first provide a quantitative analysis to motivate the need for incorporating both popularity signals from the trie and personalization signals from session queries for effective QAC. Then, we create two tries: MPC_{Main} and MPC_{Synth} for short and unseen prefixes, respectively. Finally, we explore the Retrieval-Augmented Generation (RAG; [LPP+20]) type of framework to augment top completions from both tries and fine-tune a pre-trained language model. To the best of our knowledge, this is the first attempt of trie knowledge augmentation in NLG models for personalized QAC. We have achieved state-of-the-art performance on two real *prefix-to-query* click behavior QAC datasets from Bing and AOL. On average, our model achieved a huge 57.01% and 14.33% boost in Mean Reciprocal Rank (MRR) compared to the popular trie-based lookup and the strong BART-based baseline methods, respectively.

3. We take a step towards the generation of low-resource languages across four NLG tasks with limited data/supervision. Specifically, we propose a novel unsupervised cross-lingual framework - *ZmBART* [MDKD21b]. ZmBART is developed on top of the mBART pre-trained language model and does not require parallel data/pseudo-parallel or back-translated data. This framework employs (1) intermediate unsupervised adaptive training, (2) freezing the model component (inspired by continual learning approaches), and (3) adding language tags. These measures mitigate the catastrophic forgetting problems and generate well-formed zero-shot text in LRLs. Adaptive unsupervised training is done with novel auxiliary task that requires only small monolingual data from LRLs. Here, we consider four NLG tasks and three typologically diverse languages. The proposed approach is scalable to multiple NLG tasks (as the model does not modify any hyper-parameter values across the tasks) and LRLs (operates in a zero-shot setting). Additionally, we have also created HiDG, a high-quality distractor generation dataset for the Hindi language.

4. We address the issue of non-uniform supervision transfer in cross-lingual modeling, aiming to alleviate limited supervision issues. Towards this, we propose a novel cross-lingual transfer and generation framework, Meta-X_{NLG} [MD22], based on *Model-Agnostic Meta-Learning (MAML)*, and *language clustering*. In Meta-X_{NLG} , we first cluster languages and identify the centroid language for each cluster. Subsequently, the MAML algorithm is trained using centroid languages and evaluated with non-centroid languages in a zero-shot setting. Training with a single centroid language facilitates *intra-cluster* generalization, while training with multiple centroid languages enables *inter-cluster* generalization. This way, the proposed approach exhibits more uniform cross-lingual transfer. The framework is developed on top of the mBART model. It is the first attempt, to the best of our knowledge, to explore meta-learning techniques for cross-lingual NLG. We evaluate the model's performance across two NLG tasks, 30 LRLs, and 5 popular datasets. The proposed model outperforms all strong baselines, achieving an average improvement of 13.16% in the ROUGE-L for abstraction text summarization and 7.86% in the BLEU for question generation over the strong baseline across considered LRLs and datasets.

5. We enable NLG, specifically machine translation technology, for extremely low-resource languages (ELRLs). Specifically, we have addressed the task of *ELRLs to English* machine translation (MT) by utilizing surface-level lexical similar-

ity between *closely related* ELRLs and HRL. There are many ELRLs that are lexically similar to HRLs; for example, Bhojpuri is lexically similar to Hindi. The training instance for ELRL (say, E_i) is a noisy version of the related HRL instance (say, H_i). In other words, $E_i = \eta(H_i)$ where η is noise function. We propose two novel noise augmentation strategies and apply them to the source side (i.e., HRL) of HRL-to-English parallel data to obtain noisy proxy training data for the ELRL-to-English MT task. The noise augmentation in HRL improves lexical similarity between HRL and ELRLs. We learn the vocabulary, train a stranded transformer neural network with this augmented data, and perform the evaluation in a zero-shot setting with ELRLs. Noise augmentation acts as a regularizer to account for lexical variances between HRL and ELRLs and improve cross-lingual transfer.

We have proposed two novel noise augmentation approaches: (i) CharSpan [MKDK24]: This approach randomly augments character span noise and does not require any training resources in ELRLs other than alphabets. (ii) SelectNoise [BMD23]: In this approach, noise augmentation character candidates are extracted with Byte Pair Encoding (BPE) merge operations and edit operations. Sampling algorithms are then used for noise augmentation. This approach is systematic and linguistically inspired but requires small monolingual data (1000 examples) in ELRLs. These models are evaluated with multiple ELRLs across different typologically diverse language families. Across all ELRLs and families, the CharSpan and SelectNoise models achieved CharF gains of 9.46% and 11.31%, respectively, over the vanilla neural machine translation model.

1.3 Thesis Outline

This thesis extends generative NLP in two aspects: (1) applying NLG techniques to two non-mainstream yet important NLG applications, namely, distraction generation and personalized query auto-completions, and (2) extending NLG techniques to the generation of text in low-resource languages. Considering these aspects, this thesis is divided into two parts: advancing the frontier of NLG with constraints (Chapters 3 and 4) and enabling low-resource language generation (Chapters 5, 6, and 7). The pictorial outline of the thesis is presented in Fig. 1.2 and is organized as follows:

- In Chapter 2, we provide the details of the background by introducing all the basic concepts required to understand the work presented in this thesis. Par-

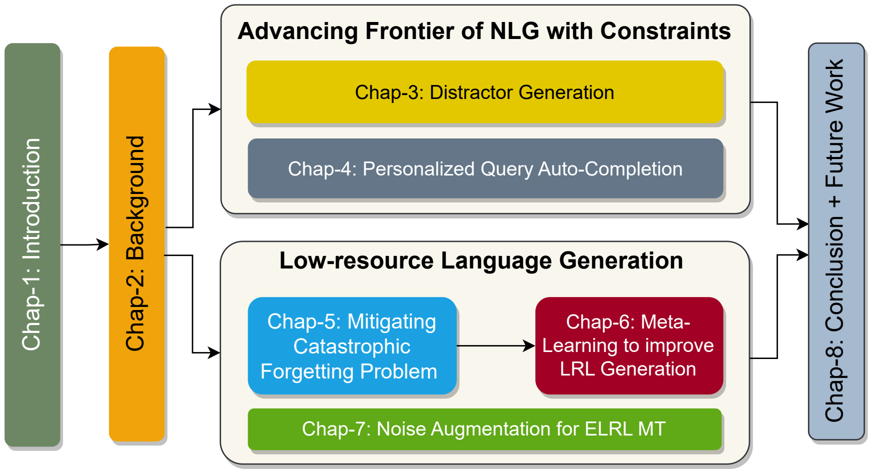

Chap-1: Introduction
Chap-2: Background
Advancing Frontier of NLG with Constraints
Chap-3: Distractor Generation
Chap-4: Personalized Query Auto-Completion
Low-resource Language Generation
Chap-5: Mitigating Catastrophic Forgetting Problem
Chap-6: Meta-Learning to improve LRL Generation
Chap-7: Noise Augmentation for ELRL MT
Chap-8: Conclusion + Future Work

Figure 1.2: Outline of the Thesis

ticularly, we introduce relevant NLG tasks, RNN and transformer architecture, neural networks for NLG, language models, pre-training, and relevant pre-trained models.

- **Advancing the Frontier of NLG with Constraints :** This part of the thesis has the following two chapters:
  - In Chapter 3, we describe a modeling approach that decouples the passage sentences and explores a multi-decoder technique to generate long, coherent, and diverse distractors in the distractor generation task.
  - In Chapter 4, we first motivate the problem of short and unseen prefixes in PQAC, supported by quantitative analyses. We conclude that limited context is the major reason for performance degradation. Subsequently, we introduced a modeling framework inspired by Retrieval-Augmented Generation (RAG). Within this framework, we harnessed trie context augmentation to address limited context problems and enhance the generation of high-quality completions.
- **Low-resource Language Generation:** This part of the thesis has the following three chapters:
  - In Chapter 5, we made our first effort in low-resource language generation. This chapter deals with mitigating the catastrophic forgetting problem and enabling well-formed zero-shot generation in low-resource languages across four NLG tasks and three languages.

  - Chapter 6 presents an advancement over Chapter 5, where we employ a meta-learning, model-agnostic meta-learning (MAML; [FAL17]), approach to enhance cross-lingual transfer for low-resource languages across 30 LRLs, two NLG tasks, and five public datasets.
  - In Chapter 7, we extend language technology to extremely low-resource languages (ELRL) by developing zero-shot machine translation systems for the ELRL to English direction. Here, we proposed two noise augmentation approaches (CharSpan and SelectNoise) to enhance cross-lingual transfer from closely related HRL and other ELRLs.

- In Chapter 8, we present important conclusions and highlight some interesting future research directions.

Chapter 2

Background

2.1 Introduction to Natural Language Generation

In this chapter, I will present the fundamental concepts necessary to understand the thesis. This chapter offers a concise technical overview of relevant natural language generation (NLG) tasks, Transformer architecture, various NLG architectures, pre-trained language models (PLMs), and multilingual PLMs. These concepts will be used in the subsequent chapters.

2.1.1 Defining Natural Language Generation Task

The natural language generation (or text generation) task can be framed as *generating human-like output text y given input x.* The input x can be text (e.g., in abstractive text summarization task), a text-tuple (e.g., in question generation task), an image (e.g., in image captioning task), or a multi-modal input (e.g., in dialogue systems where the inputs consist of both images and conversation history). Some NLG tasks, like query auto-completion, involve continuing the generation from the input text/prompt. Formally, for a given input $y_1, y_2, \ldots, y_t$, the generation continues as $y_{t+1} \ldots y_{|y|}$.

Zero-shot Generation: It is the ability of an NLG model to generate output text in a language L (or domain D) without prior explicit labeled training in L (or D).
Few-shot Generation: It is the ability of an NLG model to generate output text in a language L (or domain D) with limited labeled training examples N in L (or D). Here $N << M$, where M is the total number of examples in training data.

2.1.2 Relevant NLG Tasks

In this thesis, we have considered six NLG tasks, *viz.*, distractor generation, query auto-completions, abstraction text summarization, question generation, news headline generation, and machine translation. Details for each of the tasks are provided below:

Distractor Generation (DG)

Given a reading comprehension MCQ, i.e., `<passage, question, correct answer>` *triplet, it is the task of generating multiple incorrect options, i.e., distractors.* The distractors should be coherent, grammatically correct, and confusing. There can be multiple correct distractors for an input triplet. The ideal distractors should be semantically related (but not semantically equivalent) to the correct answer and in the context of the question. Table 2.1 presents a sample input triplet and three distractors.

Passage	Ole bull was a very famous violinist from norway. He really liked to play the violin. But his father thought that playing the violin was not useful. So his father sent him to university to study. However, playing the violin was his dream. He did n't want to give up his dream. So he left university before he finished his studies and spent all his time and energy practicing the violin. Unfortunately, his violin teacher was not very good. So when it was time for him to start his concert tour, he still couldn't play the violin very well. Therefore, a milan newspaper critic criticized him and said that he was an untrained violinist. When facing this kind of problem, some people may become very angry and some people try to learn from it. Fortunately, ole bull belonged to the second group. He went to the newspaper office and found the critic. Instead of being angry, he talked about his mistakes with the man and listened to the man's advice. After he met the critic, he gave up the rest of his concerts. Then he went back to practice the violin with the help of good teachers. In the end, he got great success when he was only 26. He also became one of the most famous violinists in the world.
Question	Why didn't ole bull's father like him to play the piano?
Correct Answer	Because he thought playing the violin was useless.
Distractor - I	Because playing the violin would cost lots of money.
Distractor - II	Because the violin was not good.
Distractor - III	Because he didn't like to play the violin.

Table 2.1: Sample triplet and corresponding distractors from Race dataset [LXL$^+$17] for DG task.

Query Auto Completions (QAC)

Let us consider a user is interacting with a search engine and has entered the previous n search queries in a *session*. Currently, the user is typing the *prefix* of a search query. *The QAC is a task of generating top-k completions, given session and prefix.* A sample session, prefix, and completions are shown in Table 2.2.

Session	hurricane resistant || hurricane lines || houston crap || houston crap plan || hurricane climate
Prefix	hurricane climate actio
Completion 1	houston climate action plan
Completion 2	houston tx climate action plan
Completion 3	houston climate controlled storage
Completion 4	houston climate control storage
Completion 5	houston climate today

Table 2.2: Sample session, prefix, and top-5 completions from Bing search engine for QAC task. '||' is search query separator in session.

Abstraction Text Summarization (ATS)

Given an input document, the task is to generate an abstractive human-like summary. The summaries are expected to be coherent, concise, grammatically correct, and to faithfully represent the information from the document. A sample input document and headline summary are shown in Table 2.3.

Document	Police were called to the scene outside the Coral shop on Compton Road in Harehills just before 14:00 BST. The man was taken to hospital for treatment but his condition is not known. West Yorkshire Police said the area has been cordoned off and officers remain at the scene. The force has appealed for information.
Summary	A man has been stabbed in broad daylight outside a betting shop in Leeds.

Table 2.3: Sample document and corresponding summary from XL-Sum dataset [HBI+21] for ATS task

News Headline Generation (NHG)

This task is closely related to the ATS task. *Given input news articles, the task is to generate concise, grammatically coherent, semantically correct, and abstractive human-like headline.* A sample input news article and the corresponding headline are shown in Table 2.4.

News Article	Scientists have discovered a new species of butterfly in the Amazon rainforest. The butterfly exhibits vibrant colors and unique wing patterns, making it a significant find for biodiversity researchers.
Headline	New Butterfly Species Discovered in Amazon Rainforest

Table 2.4: Sample news article and corresponding headline from PENS dataset [AWL+21] for NHG task.

Question Generation (QG)

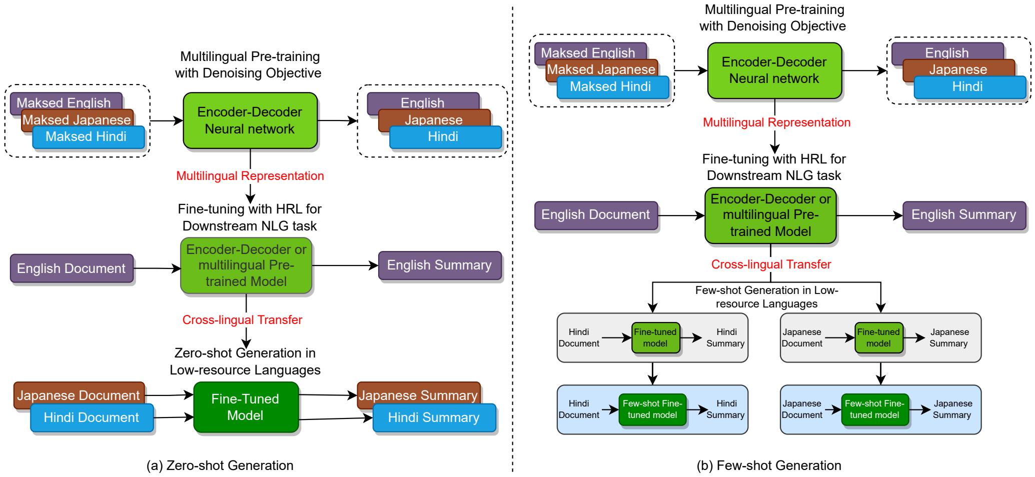
Given an input passage and correct answer, the task is to generate semantically and syntactically correct questions that can produce the answer. Any question-and-answering (Q&A) data can we used for question-generation tasks. Sample passage, answer, and question are shown in Table 2.5.

Passage	The Joan B. Kroc Institute for International Peace Studies at the University of Notre Dame is dedicated to research, education and outreach on the causes of violent conflict and the conditions for sustainable peace. It offers PhD, Master's, and undergraduate degrees in peace studies. It was founded in 1986 through the donations of Joan B. Kroc, the widow of McDonald's owner Ray Kroc. The institute was inspired by the vision of the Rev. Theodore M. Hesburgh CSC, President Emeritus of the University of Notre Dame. The institute has contributed to international policy discussions about peace-building practices.
Answer	President Emeritus of the University of Notre Dame
Question	What is the title of Notre Dame's Theodore Hesburgh?

Table 2.5: Sample passage, answer and question from SQuAD dataset [RZLL16a] for QG task

Low-Resource Language Generation

It is the task of generating textual output from a given task with a limited amount of training data or linguistic resources. In this thesis, we have considered those low-resource languages (LRLs) for which the learning data or resources are limited. In this scenario, typical learning is enabled through other high-resource languages (HRL) via cross-lingual transfer. Generations are performed in a zero-shot and few-shot setting. The low-resource language generation processes (zero-shot and few-shot) are depicted in Fig. 2.1.


Multilingual Pre-training with Denoising Objective
Maksed English
Maksed Japanese
Maksed Hindi
Encoder-Decoder Neural network
English
Japanese
Hindi
Multilingual Representation
Fine-tuning with HRL for Downstream NLG task
English Document
Encoder-Decoder or multilingual Pre-trained Model
English Summary
Cross-lingual Transfer
Zero-shot Generation in Low-resource Languages
Japanese Document
Hindi Document
Fine-Tuned Model
Japanese Summary
Hindi Summary
Multilingual Pre-training with Denoising Objective
Maksed English
Maksed Japanese
Maksed Hindi
Encoder-Decoder Neural network
English
Japanese
Hindi
Multilingual Representation
Fine-tuning with HRL for Downstream NLG task
English Document
Encoder-Decoder or multilingual Pre-trained Model
English Summary
Cross-lingual Transfer
Few-shot Generation in Low-resource Languages
Hindi Document
Fine-tuned model
Hindi Summary
Japanese Document
Fine-tuned model
Japanese Summary
Hindi Document
Few-shot Fine-tuned model
Hindi Summary
Japanese Document
Few-shot Fine-tuned model
Japanese Summary

(a) Zero-shot Generation

(b) Few-shot Generation

Figure 2.1: Illustration of zero-shot and few-shot process for low-resource language generation. Multilingual representation and Cross-lingual transfer help in LRL generation. Here, we consider abstractive text summarization as an example NLG task.

Cross-Lingual Transfer and Generation

Following Vu et al. [VBL+22], cross-lingual transfer and generation[1] is a task in which a model learns a generative task from labeled data in one language (typically English) and then performs the equivalent generative task in another language. Cross-lingual transfer facilitates low-resource language generation and is often referred to as cross-lingual generation. A sample cross-lingual generation process is depicted in Fig. 2.1.

2.1.3 Evaluation Metrics

Now, we briefly discuss the multiple automated and human evaluation metrics that are popular in the literature. These metrics have been used in this thesis to evaluate different models across various tasks. The multilingual variants of each automated evaluation metric are obtained by modifying the corresponding tokenizer, stemmer, and so on.

Automated Evalution Metrics

For each of them, we assume the generated text is evaluated with one or more reference texts. This excludes the MRR and $BLEU_{RR}$ metrics, which are information retrieval metrics. These two metrics focus on evaluating k-ranked generations, comparing them against a single reference. Below is a brief overview of each of these metrics:

- **Bilingual Evaluation Understudy (BLEU; [PRWZ02a]):** BLEU is a precision-based evaluation metric that measures the exact lexical match between a machine-generated text and human reference text. It calculates the percentage of overlapping n-grams between the generation and the reference. As this is a precision-oriented metric, to have meaningful scores for short generations, a brevity penalty is added in the BLUE score computation.

- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE; [Lin04a]):** ROUGE is a family of metrics used for evaluating the quality of the generated text, particularly in abstraction generation applications like text summarization and headline generation. It is a lexical overlap-based metric. ROUGE measures the recall of n-grams and word sequences in the generated text with respect to reference summaries.

[1] In this thesis, we use the term *cross-lingual transfer and generation* and *cross-lingual generation* interchangeably.

- **Metric for Evaluation of Translation with Explicit ORdering (METEOR; [LD09]):** METEOR is an improvement over the BLEU metric, as it allows for a flexible (approximate) match, unlike BLEU, which requires an exact match. For flexible matching, METEOR considers a combination of factors, such as unigram precision, recall, stemming, synonymy, and word order alignment, focusing on the explicit ordering of words.

- **BERTScore [ZKW+20]:** BERTScore is a metric that utilizes pre-trained BERT models [DCLT18] to evaluate the quality of generated text. It measures the similarity between token embeddings in the generated text and the reference text based on contextual information.

- **Character F-score (ChrF; [Pop15]):** ChrF is a character-level evaluation metric to assess machine translation quality. It considers character overlap, edit distance, and character n-grams to assess the quality of translation between the generated and reference translations.

- **Bilingual Evaluation Understudy with Representations from Transformers (BLEURT; [SDP20]):** It is a learned evaluation metric based on BERT representation and a few thousand human judgments. Primarily introduced for machine translation tasks but extended to many NLG tasks. It is also viewed as an extension of the BLEU metric.

- **Cross-lingual Optimized Metric for Evaluation of Translation (COMET; [RSFL20]):** It is an adaptable MT evaluation metric based on cross-lingual pre-trained language modeling that exploits information from both the source and a reference target to predict MT quality accurately. It is currently a default choice for the MT task.

- **Mean Reciprocal Rank (MRR; [SMR08]):** MRR is an information retrieval metric that calculates the mean of the reciprocal ranks of the first match of reference query with ranked generated queries. It is commonly used to assess the effectiveness of search and ranking systems.

- **BLEU Reciprocal Rank ($BLEU_{RR}$; [YSH+21]):** It is an extension of MRR metric to relax exact match as done in MRR. It is defined as weighted MRR, where the weights are the BLEU score between the reference query and generated queries.

Human Evaluation Metrics

There are two primary human evaluation methods for assessing machine-generated text quality: *direct evaluation* and *relative evaluation.* In direct evaluation, evaluators independently rate the generated text from different models. In relative evaluation, two or more generations from different models are compared and ranked in relation to each other to determine the best models. These two evaluation methods employ the following one or more popular human evaluation metrics [KSB+22]:

- **Fluency:** This metric is quantified by measuring *how fluent the generated text is?.* In direct evaluation, the scale is 1 (indicating non-fluent) to 5 (indicating very highly fluent).

- **Relatedness:** This metric is quantified by measuring *how much of the generated text is related to input or in context with input?.* In direct evaluation, the scale is 1 (indicating not at all related) to 5 (indicating very highly related).

- **Grammatical Correctness:** This metric is quantified by measuring *how grammatically correct the generated text is?.* In direct evaluation, the scale is 1 (indicating not at all grammatical correctness) to 5 (indicating very high grammatical correctness).

- **Distractability:** This metric is quantified by measuring *How confusing distractors are?.* In direct evaluation, the scale is 1 (indicating not at all confusing) to 5 (indicating very highly confusing).

2.2 Neural Networks for NLG Tasks

Neural Network: The neural network, also known as an *Artificial Neural Network (ANN)*, is a computational model inspired by the structure and function of the human brain. It comprises interconnected nodes, or artificial neurons, organized in layers. Typically, the network consists of multiple layers, with each layer receiving input from the previous layer and forwarding its output to the next layer. The network (or model) has numerous parameters, and during the training process, these parameters (also known as weights) are updated to learn generalization features/patterns. The training process consists of two phases: *forward pass* and *backward propagation.* First, the neural network weights undergo random initialization. Subsequently, the forward pass commences, wherein input is propagated through the network, resulting in output from the final layer. Following this, the output is employed to calculate the loss (error/cost function) with the model's predictions to the actual reference. Upon computing the loss, backward propagation is performed to update (adjust) the network weights. This iterative process is repeated with a large number of examples until the model converges. The optimized model is then capable of generalizing its predictions to unseen examples, i.e., predicting close to the true target value. Neural networks are the foundation for many advanced deep learning architectures, including convolution neural networks (CNNs), recurrent neural networks (RNNs), and Transformers. Deep learning models, with their deep neural networks, have achieved remarkable success in various fields, leading to breakthroughs in tasks like image recognition, natural language technology, and autonomous decision-making. More details of these foundational concepts are presented by Goodfellow et al. [GBC16].

Sequence-to-Sequence Neural Network: Within the scope of this thesis, we have focused on a specific category of neural networks known as *sequence-to-sequence* or *encoder-decoder* neural network. A sequence-to-sequence network could refer to any model that takes a sequence as input and generates a sequence as an output. This architecture is particularly suitable for natural language generation (NLG) tasks where both the input and output are sequences. Typically, a sequence-to-sequence network comprises two key modules: the *Encoder* and the *Decoder*. The encoder module is responsible for encoding the input, and the decoder module generates the output. The encoder module has two components: an embedding layer and contextual layers. In the embedding layer, words are transformed into d-dimensional vector representations. The contextual layer takes these representations and captures how each word interacts within the input, resulting in contextual representations for all the words in the input sequence. Multiple contextual layers are stacked on top of each other to capture different aspects of the input. The decoder module has a similar architecture but generates the target output sequence in an auto-regressive manner. The output is generated by conditioning on the input context and the previously generated words. This involves a language modeling task where the probability of the current word depends on prior words and the context. The contextual representation obtained from the last layer of the encoder is employed as the context in the decoder. Although there exist Decoder-only sequence-to-sequence models like GPT3 [BMR+20], or BLOOM [SFA+22], in this thesis, we have focused only on *encoder-decoder* neural networks.

2.2.1 RNN Sequence-to-Sequence Neural Network

Recurrent Neural Network (RNN): An RNN is a type of neural network architecture designed for processing sequential data. It is particularly suitable for tasks where data has a sequential or temporal nature, as in NLP, time series analysis, and speech recognition tasks. RNNs have a recurrent connection, which allows them to maintain and update a hidden state as they process each element of a sequence. This hidden state captures information from previous elements, enabling RNNs to model dependencies and patterns in sequential data. In the backpropagation, the gradients (derivatives of the loss with respect to the model parameters) are computed and used to update the model parameters. However, if the input sequence is long, gradients can become extremely small as they are propagated backward through time. This is called a *vanishing gradient problem.*

Long Short-Term Memory (LSTM): It is a specialized type of RNN architecture that reduces the effect of vanishing gradient as in standard RNNs. It is designed to capture long-term dependencies in sequential data. LSTMs incorporate memory cells and gates that control the flow of information. They can retain and selectively update information over longer sequences, making them well-suited for tasks that involve understanding and remembering context over time.

RNN/LSTM Sequence-to-Sequence Network: It is the standard sequence-to-sequence network in which the encoder and decoder modules are replaced with an RNN or LSTM network. The last token representation from the last layer of the LSTM/RNN input is referred to as the encoder context and is utilized by the decoder during generation. To capture long-term context from the encoder, the attention module is used in the sequence-to-sequence model. It is used to improve the model's ability to capture and weigh different parts of the input sequence during the generation of the output sequence. Consequently, the model dynamically considers important aspects of the input. These models are commonly referred to as *attention-based sequence-to-sequence* models.

With these efforts, the problem of the vanishing gradient is reduced but not completely mitigated. Additionally, in RNNs, the input is processed sequentially, which inhibits parallel processing and leads to slow training. To overcome these limitations, the Transformers neural network was introduced by Vaswani et al. [VSP$^+$17], based solely on the attention mechanism. Next, we will discuss transformer architecture in detail.

2.2.2 Transformers Sequence-to-Sequence Neural Network

Transformer Architecture: Vaswani et al. [VSP+17] proposed the Transformer neural network architecture powered with *self-attention* mechanism. It is a more complex model than ANNs, RNNs, and LSTMs. *The key idea is self-attention, by which a representation at a position is computed as a weighted combination of representations from other positions.* In Transformer, the input sequence of word vectors X represents a corresponding query sequence vectors Q, key sequence vectors K, and value sequence vectors V. Each vector has dim d. The key and query at every position are compared to calculate how much attention to pay to each position (i.e., self-attention); based on this, a weighted average of the values at all positions is calculated (see Equation 2.2). This operation is repeated many times at each level of the transformer neural net, and the resulting value is further manipulated through a fully connected neural network layer and through the use of normalization layers and residual connections to produce a new vector for each word. This whole process is repeated many times, giving extra layers of depth to the transformer neural net.

$$K = W_K X, Q = W_Q X, V = W_V X \tag{2.1}$$

$$SelfAttention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2.2}$$

This architecture enables parallel input processing and mitigates vanishing gradient problems. As the input is processed parallel, the word order information is injected by adding position embeddings [VSP+17, GAG+17] with each word embedding before the self-attention operations. Due to the impressive performance in terms of modeling power and training speed, Transformer neural networks become *de-facto* in NLP modeling nowadays.

Transformers Sequence-to-Sequence Network: Here, both the encoder and decoder modules are Transformer networks. Further, the decoder employs additional attention called *encoder-decoder/cross-attention*, which selectively focuses on the encoder's context in the decoder during the generation process. The generation process with the decoder is auto-regressive in nature.

2.2.3 Tokenization Techniques

Given input text $X = \{x_1, x_2 \ldots x_{|X|}\}$, how to transform this into the embedding sequence $E = \{e_1, e_2 \ldots e_{|E|}\}$ that can be passed as input to the neural network model? In the past, a fixed vocabulary was defined, and common words were assigned word embeddings (either random or from word2vec/GloVe), while rare or out-of-vocabulary or unknown words were represented as `<UNK>` leading to a lack of word identity information. To address this issue, more successful *subword tokenization* techniques emerged. These tokenizers break the words into smaller, frequent subwords. For example, the word ‘unhappy’ can be segmented into two smaller units ‘un’ and ‘happy’. One of the popular subword segmentation techniques is Byte Pair Encoding (BPE; [SHB16b]). The idea behind BPE is to iteratively replace the most frequent pair of character n-grams in a sequence with a single, unused character n-gram. BPE allows the model to encode and generate any Unicode string as a sequence of in-vocabulary tokens; while more common words have their own embeddings, rarer words are split into smaller pieces with their own embeddings. Below are informal steps to create BPE vocabulary:

1. **Initialize with Character Vocabulary:** Start with a character-level vocabulary where each character is considered a token.

2. **Represent Input Text:** Represent input text data using this character-level vocabulary. Each word is split into a sequence of characters, and a special end-of-word symbol (e.g., ’</W>’) is added to indicate the word boundaries.

3. **Count Symbol Pairs:** Iterate through input data and count the frequency of all symbol pairs.

4. **Merge Most Frequent Pair:** Identify the most frequent pair in the input data. This pair will be merged, and replace all occurrences of this pair with the new merged symbol.

5. **Update Vocabulary:** Add the newly created merged symbol to your vocabulary.

6. **Repeat:** Repeat steps 3 to 5 for a predefined number of iterations or until a certain vocabulary size is reached.

This approach is beneficial for handling languages with complex morphology, agglutinative languages, and out-of-vocabulary words. Subword tokenization enables efficient multilingual support, improved generalization, and reduced vocabulary size, making it a fundamental tool for building versatile and memory-efficient NLP models. There are many other subword tokenization methods like sentencepeice [KR18] similar to/based on BPE. Usually, each sequence appends with two special tokens: `<START>` and `<END>`. In training RNNs or Transformers, the <START> token is typically the first input (y_1) at the initial timestep. This guides the model to learn when to stop generating by predicting the <END> token.

2.2.4 Training and Evaluation of NLG Neural Network

Maximum Likelihood Estimation (MLE) Training: As previously mentioned, training neural networks usually entails end-to-end training through forward and backward propagation, adjusting weights using optimizers such as stochastic gradient descent (SGD). The training objective, known as maximum likelihood estimation (MLE), quantifies how *well the model fits the target data.* MLE is essentially the *negative log-likelihood* of the target data based on the encoder-decoder model, which is equivalent to measuring the *cross-entropy loss* between the data distribution and the model. Formally, given training examples $(x, y) \in D_{train}$, the cross-entropy loss for an encoder-decoder model (M) is:

$$CrossEntropy(D_{train}) = -\frac{1}{n} \sum_{(x,y) \in D_{train}} logP_M(y|x) \tag{2.3}$$

Where n is the total number of tokens in the output sequences y.

Text Generation: Generating text with encoder-decoder NLG models is a two-step process. First, the encoder processes the input data to create a context vector. Second, the decoder utilizes this context vector to produce the output sequence, one token at a time, with a decoding algorithm. The choice of decoding algorithms, such as greedy, beam search, or sampling techniques, impacts how the output generation is required. This process is often referred to as "*conditional generation*" because it is conditioned on the input context.

2.3 Pre-training of Neural Network Models

In this section, we will briefly review recent advancements in various types of *transformer-based language models* and their pre-training processes. We will also discuss how these language models are adapted for multilingual setups. Finally, we will wrap up this section by mentioning the relevant language model used in this thesis.

2.3.1 Language Models

The Language Model (LM) aims to model the *generative likelihood* of word sequences in order to predict the probabilities of future (or missing) tokens. Due to the parallelization and capacity, the Transformer has become the *de-facto* backbone to develop various language models as they are scalable to hundreds or thousands of billions of parameters. The following are the three major architectures of language models:

Encoder-only Language Models: These language models utilize only the encoder part of the transformer model for language modeling. They are typically modeled using a masked language model objective and are also referred to as *Masked Language Models (MLMs)*. Their primary objective is to predict the identity of masked tokens in a token sequence y. For example, in the input sequence '*I love <MASK> book.*', the language model predicts the token that is most likely to replace <MASK>'. Formally, for a text token sequence $y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_{|y|}$, the encoder-only model calculates the probability of a masked token with encoder-only language models (P_{Enc}) as follows:

$$P_{Enc}(y_t|y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_{|y|}) \tag{2.4}$$

A few popular encoder-only models are BERT [DCLT18] and ROBERTA [LOG+19].

Decoder-only Language Models: These language models generate the next token (or continuations of text) of input text sequence or assign a probability $P_{Dec}(y)$ to a sequence of text y (e.g., for ranking task). These models are also called as *Causal Language Models (CLM)*. The probability of sequence can computed using the chain rule with decoder-only language models (P_{Dec}) as follows:

$$P_{Dec}(y_1, y_2, \ldots y_t) = \Pi_{i=1}^{t} P_{Dec}(y_i|y_1, y_2, \ldots, y_{t-1}) \tag{2.5}$$

A few popular decoder-only language models are GPT [RNS+18, BMR+20] and BLOOM [SFA+22].

Encoder-Decoder Language Models: These language models assign probability $P(y|x)$ to a text sequence y given some input x. These language models consist of both encoder and decoder modules. The encoder adapts stacked multi-head self-attention layers to encode the input sequence for generating its latent representations while the decoder performs cross-attention on these representations and *auto-regressively* generates the target sequence. These encoder-decoder language models have denoising objectives where the modeling has to generate the correct input sequence from a noisy input. It is called denoising auto-encoding (DAE). The noise can be token masking, span masking, sentence order permutation, and so on. Few popular architectures are BART [LLG+19] and T5 [RSR+20a]. In DAE modeling, the generated correct output text y is conditioned on some noisy/corrupted input x or computes the conditional probability of a given (x, y) pair with encoder-decoder language models language model P_{EncDec} as:

$$P_{EncDec}(y_1, y_2, \dots y_t|x) = \Pi_{i=1}^{t} p_{EncDec}(y_i|y_1, y_2, \dots y_{t-1}, x) \tag{2.6}$$

All three language models can also be used to obtain the embedding representation for input sequences, which can be further utilized in downstream NLP applications. This representation serves as a powerful warm-up or starting point for any neural network. Decoder-only LMs predict the next token based on the left context (i.e., auto-regressive), while encoder-only LMs use bidirectional context for prediction. The encoder-decoder LMs have both bidirectional context and auto-regressive generation, making them more effective and suitable for NLG tasks. Moreover, masked language models are not generally intended or used to generate text, with some exceptions. Considering this, in this thesis, we have used only encoder-decoder-based language models.

2.3.2 Pre-training and Fine-Tuning of Language Models

The remarkable modeling capabilities of Transformer networks are indeed exciting; however, the training of these neural networks depends heavily on the availability of large annotated datasets. This presents a challenge for downstream applications and low-resource languages where the annotated data is limited. To address this challenge, pre-training-based modeling was emerged as a hopeful direction and currently a dominating paradigm in NLP. This paradigm has two phases: *pre-training* and *fine-tuning*.

- **Model Pre-training:** In the pre-training phase, a language model (with a large number of parameters) is trained on a large text corpus, consuming thousands of GPU hours. The core objective of pre-training is to enable the model to learn statistical patterns, relationships, and data representations. It involves training the model with *denoising auto-encoding* (DAE) objectives, i.e., predicting missing words, sentence permutation, and so on. Pre-training is unsupervised and often referred to as *self-supervised* training.

- **Fine-Tuning:** After the pre-training phase, the model undergoes fine-tuning for a specific downstream task. Fine-tuning is the process of adapting the pre-trained model to a task with a minimal number of training examples. It is a form of supervised learning as it relies on labeled data. The fine-tuning process capitalizes on the general language understanding acquired during pre-training to enhance performance on the particular task. This knowledge transfer (also known as *Transfer Learning*) often results in improved performance even with limited amounts of task-specific data.

2.3.3 Multilingual Pre-training of Language Models

In this thesis, we have mostly utilized the multilingual variants of encoder-decoder language models. Therefore, here, we will restrict our discussion to only encoder-decoder LMs in a multilingual context. The underlying model architecture for most multilingual LMs is similar to the base (English) LMs. The training of multilingual LM extends the *denoising auto-encoding (DAE)* objective in multilingual setup as follows:

Multilingual Denoising Auto-Encoding (mDAE): Given monolingual text data covers K languages: $D = D_1, D_2, \ldots, D_K$ where each D_i is a collection of monolingual documents in language i. We train an encoder-decoder language model to predict the original text X given $\eta(X)$, where η is the noise function. While training, we minimize mDAE loss ($\mathcal{L}_\theta$) as:

$$\mathcal{L}_\theta = \sum_{D_i \in D} \sum_{X \in D_i} logP_{EncDec}(X|\eta(X); \theta) \tag{2.7}$$

where θ is model's learning parameter and X is an instance in language i. For each instance of a batch, a random language is sampled, and corresponding language sentences are packed until it reach the document boundary or reaches the maximum token length limit. The language model is trained with this batch using a DAE self-supervised training objective. This process is repeated for many batches and epochs until the model's perplexity converges. This will allow the model to represent many languages in a common latent representation space. Few popular models are mT5 [XCR+21] and mBART [LGG+20d]. It is demonstrated [HRS+20] that these multilingual pre-trained language models are effective in many downstream applications in multiple languages. Additionally, this drives the cross-lingual modeling to enable zero-shot and few-shot evaluation/generation.

2.3.4 Relevant Pre-trained Language Models

In this thesis, we have used the following pre-trained language model for multiple NLG tasks:

- **GPT2 [RNS+18]:** GPT2 is the decoder-only language model that is trained with the next token prediction objective in an auto-regressive manner. This is the underlying objective for large models like GPT3 [BMR+20] and BLOOM [SFA+22]. The model has different checkpoints depending on the number of model parameters/layers.
- **BART [LLG+19]:** BART is a pretraining sequence-to-sequence model trained with a denoising autoencoder objective. It is trained using corrupted text generated by an arbitrary noising function and learns to reconstruct the original text. The base version of the model consists of 6 layers of transformer encoders and 6 layers of decoders. It is evaluated across a wide range of NLG tasks, including abstractive text summarization and dialogue response generation, among others.
- **mBART [LGG+20d]:** It is a multilingual version of the BART model with a multilingual DAE objective. The model has two versions, one with 25 languages and the other with 50 languages. The model was evaluated with the machine translation task.
- **mT5 [XCR+21]:** It is multilingual version of the T5 [RSR+20a] pre-trained sequence-to-sequence model. The model is pre-trained with 101 languages. T5 has a DAE objective in which the model predicts missing input tokens/spans. T5 is designed to handle a wide range of NLP tasks in a unified manner. The key idea behind T5 is to cast all NLP tasks into a text-to-text format, where both input and output are represented as sequences of text. This allows the use the same model, loss function, hyperparameters, etc., across a diverse set of tasks.

Chapter 3

Advancing Frontiers of NLG: Distractor Generation

3.1 Introduction

This chapter presents our first efforts in the natural language generation with traditional Long Short-Term Memory (LSTM)-based [HS97] modeling for *diverse text generation* within the Distractor Generation (DG) task. This work was carried out in a time span when the LSTM-based deep learning models were the preferred choice over emerging Transformers models. First, let us understand the distractor generation task and associated challenges.

Reading comprehension (RC) is recognized as an advanced cognitive task in NLP, which involves both shallow and deep understanding of articles to carry out complex inferences. A person can demonstrate his/her understanding of an article by answering questions about the article. Particularly, multiple-choice questions reading comprehension (*RC-MCQ*) is a popular assessment technique to judge human understanding. It provides several advantages, including fast, unbiased, quick, and consistent evaluation. In the classical convention, MCQ consists of a triplet: (1) question, (2) correct answer, and (3) *distractors* or the incorrect answers to confuse examinees [CS18]. Out of the three components, the creation of high-quality distractors is an important, challenging, and time-consuming task [WLG17]. According to Goodrich et al. [Goo77], *ideal distractors should be semantically related (but not semantically equivalent) to the correct answer and in the context of the question.* Therefore, automation of the distractor generation process is challenging but, at the same time, beneficial to target audiences. We aim to leverage LSTM-based deep learning models to generate long, grammatically correct, and non-obvious/confusing distractors.

The distractor generation system can be utilized for educational purposes in language learning assessment. As a reverse task, the system can also be used to automatically create annotated datasets to push research in reading comprehension and Question and answering (Q&A) tasks. Additionally, the variant of distractor generation models can be used for many other NLG applications: (1) generating diverse utterances for a monotonous input/state in conversational systems, (2) generating diverse headlines for an input article to attract a large audience and many more.

In order to generate good distractors that are semantically correct but not equivalent to the correct answer, we try to understand how humans usually extract distractors. According to our understanding, humans generally follow a two-step generation process - (a) *search for article sentences that are in context with the question* and (b) *avoid sentences that are semantically equivalent to the answer*. The resultant sentences are potential candidates for distractor generation. Inspired by the human approach, we adapted a data-driven, sequence-to-sequence learning framework to address the problem of automated distractor generation. We propose a novel hierarchical multi-decoder network (**HMD-Net**). In HMD-Net, we first obtain the contextual word-level and sentence-level representations of the article by a hierarchical encoder. Additionally, word-level representations are learned for question and answer separately. Then, we use *SoftSel* operation and *Gated Mechanism* to capture rich semantic relations among these components. On the decoder side, we used three different decoders with a dis-similarity loss to generate three distractors. Three decoders learn to generate three diverse distractors from candidate sentence(s) so that they have similar contexts but are not the same.

We carefully reviewed the generated text from HMD-Net and observed that there are sentences that have gender errors, morphological errors, etc. A few examples are "*She is good at solving maths.*" where, based on the context, the correct sentence should be: "***He** is good at solving maths.*" and "*Mr. Robert went last months.*" where the correct sentence is: "*Mr. Robert went last **month**.*" This indicates that the model sometimes fails to learn the linguistic properties of the word. To eliminate such problems, we externally included linguistic feature representation along with word representation in HMD-Net as used in the task of machine translation in [SH16]. Finally, to capture the contextual representation of words, we leveraged the representation from the BERT [DCLT18] model. We evaluated our system on two datasets, RACE DG [GBL+19] and RACE++ DG dataset (prepared by us) across seven automated (word-overlap based: BLEU, ROUGE, and METEOR and embedding-based) metrics and two manual evaluation metrics (grammatical correctness and distractibility). Additionally, we checked how confusing the generated distractors are by performing the human assessment. Our extensive experimentation exhibits that our model consistently outperformed all the previous baselines and emerged as a state-of-the-art model.

Our key contributions with this work are listed below:

1. We propose a novel **HMD-Net** [MD20a]**:** *Hierarchical Multi-Decoder Network* to tackle the task of automated distractor generation. It is an end-to-end, data-driven model to generate three diverse distractors from three decoders.

2. We utilize *SoftSel* operation and *Gated Mechanism* to ensure the generated distractors are in context with questions but not semantically similar to the correct answer.

3. We introduce a novel dis-similarity loss in HMD-Net for distractor generation and a new BERT cosine similarity (BERT-CS) based metric for automated evaluation.

4. We release a new high-quality distractor generation dataset RACE++ DG [1], prepared from RACE++ dataset by leveraging contextual similarities among the different components of the data instances by using a state-of-the-art contextual representation - BERT model. We conduct further analyses and evaluations to ensure the quality of the dataset.

3.2 Related Work

The task of automated distractor generation (DG) is aligned with the multiple choice question (MCQ) generation research direction. Traditionally, rule/heuristic-based distractor generation models use approaches like different similarity measures, ontology, and embedding for distractor selection [CS18]. However, in almost all cases, the granularity of generated distractors is limited to word-level or phrase-level. With advancements in deep learning, the generation of long and coherent distractors using learning-based approaches is receiving a lot of attention. A brief overview of traditional and deep learning-based approaches for automated distractor generation is presented below:

[1] code and data link: https://github.com/kaushal0494/HMD_Network

3.2.1 Traditional Approaches

Traditional methods for DG used linguistic resources like WordNet [M+03] and Thesaurus [SSY05] for determining conceptual similarity and used that information to generate the distractors. Later, linguistic properties like morphological and phonetic similarities [PE09], n-gram co-occurrence [HS16], and context similarities [PHE08] were used for extracting distractors. More popular traditional approaches use embedding-based similarities [GKK+16, JL17] between text representations obtained using GloVe, word2vec, and so on. Zesch et al. [ZM14] proposed a popular two-step process: (1) compute the ranking of potential candidate texts by a weighted combination of different similarity metrics and (2) check the reliability of candidate distractors using contextual information. However, with these models, the granularity of generated distractors is limited to word-level or phrase-level - limiting their practical uses.

3.2.2 Learning Based Approaches

Early neural approaches for DG were oriented towards a *learning-to-rank* framework. Few popular approaches [SAK13, LYW+17] viewed the distractor generation problem as a multi-class classification problem. The model proposed in [LYW+17] learns distractor-distribution conditioned on the question using generative adversarial nets (GANs). A method to generate distractors for fill-in-the-blank questions was proposed in [SAK13]. Liang et al. [LYD+18] used feature-based ensemble and neural net-based models to rank distractors. Two recent works close to our line of research are presented in [GBL+19] and [ZLW20]. Gao et al. [GBL+19] focuses on static and dynamic attention from a hierarchical encoder-decoder model. Static attention helps to identify candidate sentences from the article, and dynamic attention is then used to generate distractors. In an improvement over this, Zhou et al. [ZLW20] exploits information across articles and questions using the co-attention mechanism. They apply Jaccard Similarity (JS) to sample three distractors over a pool of distractors generated by beam search. The Jaccard similarity-based distractors sampling leads to distractors that are different at the lexical level, but either they are not in context with the question or too obvious for the end-user to eliminate. There were no precautions taken by [ZLW20] to ensure that generated distractors should not be answer-revealing or semantically equivalent to the answer as they did not consider answer text in the model. Our novel framework mitigates these limitations and generates long, robust, and confusing distractors.

3.3 Problem Statement

In automated distractor generation, we aim to generate long, coherent, grammatically correct, and confusing wrong options given a triplet `<article/passage, question, correct answer>`. Generated distractors should be in the context with the question but should not be semantically equivalent to the answer. Formally, let $S = \langle s_1, s_2, \dots s_p \rangle$ denote the input article/passage with p sentences; Each sentence s_i is word sequence of length k i.e., $s_i = \langle w_{i,1}, w_{i,2}, \dots w_{i,k} \rangle$. The question is denoted by Q and is a sequence of n words, $Q = \langle q_1, q_2, \dots q_n \rangle$. The word sequence $A = \langle a_1, a_2, \dots a_l \rangle$ of length l denotes the answer. The three distractors are represented as $D_i = \langle d_{i,1}, d_{i,2}, \dots d_{i,u_i} \rangle$ for $i = \{1, 2, 3\}$ where u_i is the length of i^{th} distractor. Our goal is to generate D_1, D_2, and D_3 given the triplet $\langle S, Q, A \rangle$.

$$D_i = \arg\max_{\bar{D}_i} log\, P(\bar{D}_i | S, Q, A; \theta_i) \tag{3.1}$$

$log\, P(\bar{D}_i | S, Q, A; \theta_i)$ is conditional log-likelihood of i^{th} distractor and θ_i is parameters associated for i^{th} distractor.

3.4 Methodology

3.4.1 Model Overview

The standard LSTM-based sequence-to-sequence architecture for automated distractor generation can be a primary choice. However, these model suffers due to the large size of the input article (RACE datasets have 342 tokens/article on average). To mitigate this, the hierarchical sequence-to-sequence models were explored [GBL+19, ZLW20]. Further, these models are suitable for generating a single distractor and fail to generate multiple distractors correctly. In this paper, we propose an advancement over the hierarchical sequence-to-sequence model and add multiple decoders to overcome the stated limitations.

As our goal is to generate three diverse distractors, we view this problem as a *one-to-many mapping* modeling setup, i.e., a single encoder for input triplet and three decoders for generating three distractors. At the **Encoder Side**, we first obtain the contextual word-level and sentence-level representations of the input triplet by the hierarchical encoder. Then, we use *SoftSel* operation and *Gated Mechanism* to capture rich semantic relations among the components of the triplet. This semantic information is exploited to find relevance scores for article sentences. The scoring

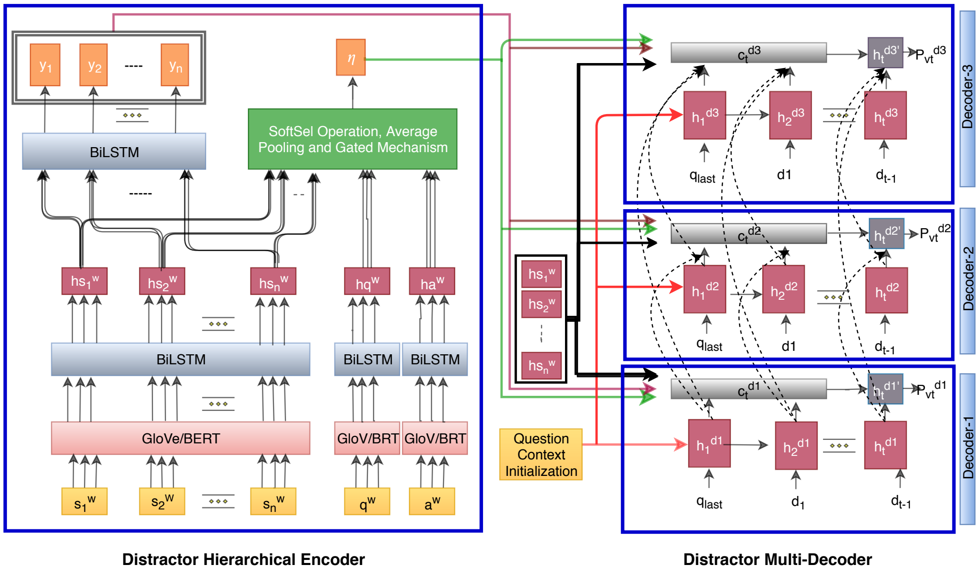


Figure 3.1: Architectural diagram of the proposed Hierarchical Multi-Decoder Model (*better viewed in color*)

decouple sentences that are in context with the question but are not semantically equivalent to the correct answer. Sentences with high relevance scores are potential candidates for distractor generation. At the **Decoder Side**, we employed a multi-decoder model with a combination of cross-entropy and dis-similarity loss to generate three distractors. This novel architecture is trained in an end-to-end manner to generate high-quality diverse distractors. The training datasets consist of less than three distractors (∼2.1 for RACE and ∼2.3 for RACE++) for many input triplets (see Table 3.1), which poses additional challenges. Hence, we consider each training example as 4-tuples, i.e., `<article, question, correct answer, distractor>` where `<article, question, correct answer>` is input and `<distractor>` is target. For instance, if a triple has two distractors, then this forms two separate training examples. During inference, the model generates three distractors for each input triplet. Fig. 3.1 presents an architectural overview of the proposed HMD-Net. We now present a detailed description of each component of our proposed HMD-Net model.

3.4.2 Hierarchical Encoder

In this section, we describe the different components of the encoder. The flow diagram of the hierarchical encoder is shown in Fig. 3.2.

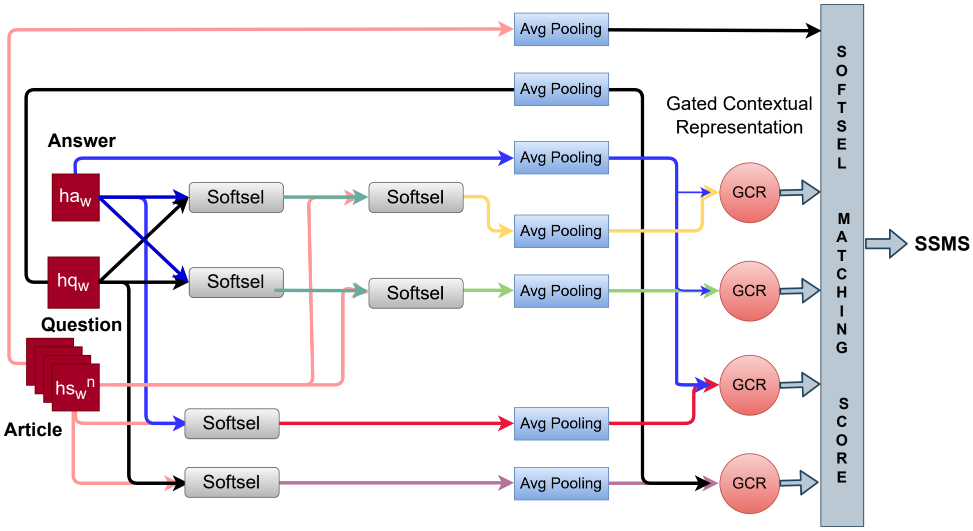


Figure 3.2: Flow diagram of Hierarchical Encoder (*better viewed in color*)

Input Word Embedding

We obtain word embedding for each component of the input triplet in two different ways.

1. We use pre-trained GloVe embedding to map words/tokens to vector representations. In addition, four linguistic feature representations are concatenated with each token. These features are: *Parts of Speech tags (f_1)*, *Named Entities (f_2)*, *root form (lemma) of the word (f_3)* and *Dependency Parsing Labels (f_4)* extracted from *Stanford CoreNLP* package.

   - Sentence token embedding $se_{i,j} = [GloVe(w_{i,j}); f_{1_{i,j}}; f_{2_{i,j}}; f_{3_{i,j}}; f_{4_{i,j}}]$
   - Question token embedding $qe_i = [GloVe(q_i); f_{1_i}; f_{2_i}; f_{3_i}; f_{4_i}]$
   - Answer token embedding $ae_i = [GloVe(a_i); f_{1_i}; f_{2_i}; f_{3_i}; f_{4_i}]$

   Here, $se_{i,j}$ indicate j^th token embedding of i^th sentence of the article. qe_i indicate i^{th} token embedding of question and ae_i indicate i^{th} token embedding of answer.

2. The BERT feature extraction method is used to obtain representation for each token. The output from the BERT model is a representation of word pieces, which are further aggregated (average pooling) to produce the final token representation.
   $se_{i,j} = BERT(w_{i,j})$, $qe_i = BERT(q_i)$ and $ae_i = BERT(a_i)$

Article Word Encoder and Sentence Encoder

The initial token embeddings $se_{i,1}, se_{i,2}, \ldots, se_{i,k}$ of article sentence s_i are fed through a Bidirectional Long Sort Term Memory (BiLSTM) network referred as $LSTM^w$ to generate contextual representations of the tokens.

$$\overrightarrow{h_{i,j}^{enc}} = \overrightarrow{LSTM^w}(\overrightarrow{h_{i,j-1}^{enc}}, se_{i,j}) \tag{3.2}$$

$$\overleftarrow{h_{i,j}^{enc}} = \overleftarrow{LSTM^w}(\overleftarrow{h_{i,j+1}^{enc}}, se_{i,j}) \tag{3.3}$$

$\overrightarrow{h_{i,j}^{enc}}$ and $\overleftarrow{h_{i,j}^{enc}}$ are forward and backward hidden representations of $LSTM^w$. The final hidden state is $h_{i,j}^{enc} = [\overrightarrow{h_{i,j}^{enc}}; \overleftarrow{h_{i,j}^{enc}}]$. We denote h_p as the sequence of hidden states $(h_{i,j}^{enc})$ of all tokens of the article.

In order to represent each sentence s_i in the article, we employed another bidirectional LSTM layer ($LSTM^s$) on top of the word encoding layer. The inputs for $LSTM^s$ are the last token (k) hidden representation of sentence s_i from $LSTM^w$ i.e., $h_{i,k}^{enc} = [\overrightarrow{h_{i,k}^{enc}}; \overleftarrow{h_{i,k}^{enc}}]$ and the first token hidden representation of sentence s_i from $LSTM^w$ i.e., $h_{i,1}^{enc} = [\overrightarrow{h_{i,1}^{enc}}; \overleftarrow{h_{i,1}^{enc}}]$. The final encoded contextual representation of i^{th} a sentence is denoted as y_i. Now, the article can be represented as $y =< y_1, y_2, \ldots y_p >$. This completes the hierarchical structure of the encoder.

Question Encoder and Answer Encoder

To determine contextual representations of the question-and-answer tokens, the initial token embeddings are fed through a bidirectional LSTM. This LSTM network is shared with article word-level LSTM.

$$h_i^q = [\overrightarrow{LSTM^w}(\overrightarrow{h_{i-1}^q}, qe_i); \overleftarrow{LSTM^w}(\overleftarrow{h_{i+1}^q}, qe_i)] \tag{3.4}$$

$$h_i^a = [\overrightarrow{LSTM^w}(\overrightarrow{h_{i-1}^a}, ae_i); \overleftarrow{LSTM^w}(\overleftarrow{h_{i+1}^a}, ae_i)] \tag{3.5}$$

The question and answer contextual representations are $h_q =< h_1^q, h_2^q, \ldots h_n^q >$ and $h_a =< h_1^a, h_2^a, \ldots h_l^a >$ respectively.

SoftSel Operation

We investigate the effect of exploiting rich interactions among the word-level representations among the components of the triplet. It turns out that these interactions are helpful in finding potential candidate sentences of the article for DG. Such in-

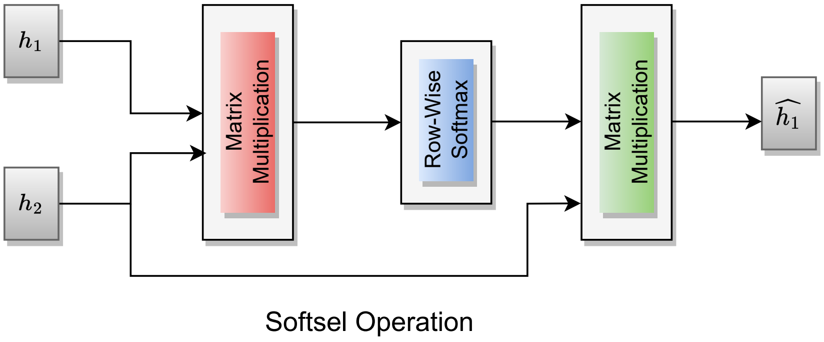


Figure 3.3: Flow diagram of softsel operation

teractions can be achieved using SoftSel operation [TCZ19], *which encodes the most relevant aspects of a sequence to another sequence.* The input to *SoftSel* operation are two sequences, and the output is an encoded sequence. A pictorial flo of softsel operation is depicted in Fig. 3.3. The operation has three steps:

1. **Cartesian Similarity:** For given two input sequences $h_1 \in \mathbb{R}^{r\times l_1}$ and $h_2 \in \mathbb{R}^{r\times l_2}$, a cartesian similarity $L \in \mathbb{R}^{l_1\times l_2}$ is obtained across all possible states or words in h_1 and h_2.
$$L = h_1^T W^L h_2 \tag{3.6}$$

2. **Row-wise Softmax:** To obtain distribution $\bar{L} \in \mathbb{R}^{l_1\times l_2}$ over cartesian similarity scores $softmax$ is applied on each row of L separately.
$$\bar{L} = \text{row-wise}\ softmax(L) \tag{3.7}$$

3. **Weighted Sum :** Finally, a weighted sum of the second sequence h_2 is encoded at the given state j of the first sequence h_1 and denoted as $\bar{h}_{1j} \in \mathbb{R}^{r\times 1}$. $\bar{h}_{1j}$ may be considered as the representation of j^{th} state of the first sequence determined from the most influential parts of the second sequence for that state.
$$\bar{h}_1 = h_2 \bar{L}^T \tag{3.8}$$

$W^L \in \mathbb{R}^{r\times r}$ is a weight matrix learned during the training process.

Evidence Encoder

We leverage *softsel* operation to encode relatedness/similarity among the components of triplet and term it as *evidence*.

- **Question-Evidence Encoder:** First, we extract *evidence* between the question and the passage. More specifically, each state of the question sequence is represented as the weighted sum of state representations of the article sequence h_p. The final encoded question sequence $\bar{h}_q$ can be viewed as a synthesized evidence vector known as the question-evidence encoder.

$$\bar{h}_q = SoftSel(h_q, h_p) \tag{3.9}$$

- **Answer-Evidence Encoder:** Similar to the question-evidence encoder, another softsel operation is performed to obtain the answer-evidence encoder $\bar{h}_a$.

$$\bar{h}_a = SoftSel(h_a, h_p) \tag{3.10}$$

- **Question-Answer-Evidence Encoder:** It is a two-step process. We first apply softsel operation between question sequence h_q and answer sequence h_a to encode the most relevant aspects of the answer in $\bar{h'}_q$. Than, another softsel operation is applied between $\bar{h'}_q$ and article hidden sequence h_p to obtain question-answer-evidence representation $\bar{h}_{qa}$.

$$\bar{h'}_q = SoftSel(h_q, h_a) \tag{3.11}$$

$$\bar{h}_{qa} = SoftSel(\bar{h'}_q, h_p) \tag{3.12}$$

- **Answer-Question-Evidence Encoder:** Similar to the question-answer-evidence encoder, another set of softsel operations are performed to obtain question-answer-evidence representation $\bar{h}_{aq}$.

$$\bar{h'}_a = SoftSel(h_a, h_q) \tag{3.13}$$

$$\bar{h}_{aq} = SoftSel(\bar{h'}_a, h_p) \tag{3.14}$$

Note that the softsel operation is not symmetric. So the representations $\bar{h}_{qa}$ and $\bar{h}_{aq}$ are different. Computation of $\bar{h}_q$ detects question-relevant sentences from the article. Whereas the other three evidence encoders are majorly oriented towards the answer and detect answer-relevant sentences in the article. This information is used in subsequent stages to ensure that the candidate sentences for distractor generation are not semantically equivalent to the answer.

Gated Contextual and Evidence Encoding Layer

Inspired by the work on a sequential variant of highway network [TCZ19], we adopted a gated mechanism (Gtd_Mchsm) to control and encode information flow between contextual representation (i.e., from LSTM network) and evidence representation (i.e., from softsel operation). Let's define the gated mechanism, which will result in a contextual-evidence representation $\bar{K} \in \mathbb{R}^{rxm}$ given contextual representation $C \in \mathbb{R}^{rxm}$ and evidence representation $E \in \mathbb{R}^{rxm}$ as given below:

$$z = \sigma(W^C C + W^E E + b) \tag{3.15}$$

$$\bar{K} = C * z + E * (1 - z) \tag{3.16}$$

The functionality of z is similar to that of the *reset gate* of RNNs. It determines what fraction of past knowledge (C) to forget and what fraction to retain. $W^C, W^E \in \mathbb{R}^{rxr}$, $b \in \mathbb{R}^r$ are parameters.

Average pooling: We applied average-pooling operation to transform contextual representations (article sentence (s_i), question (q) and answer (a)) and evidence representations ($\bar{h_q}$, $\bar{h_a}$, $\bar{h_{qa}}$ and $\bar{h_{aq}}$) to obtain fixed length vector representations.

$$\mathbf{s_i} = \frac{1}{k}\sum_{t=1}^{k} h_{i,t}^{enc}, \mathbf{q} = \frac{1}{n}\sum_{t=1}^{n} h_t^q, \mathbf{a} = \frac{1}{l}\sum_{t=1}^{l} h_t^a \tag{3.17}$$

$$\mathbf{h_q} = \frac{1}{n}\sum_{t=1}^{n} \bar{h}_{q_t}, \mathbf{h_a} = \frac{1}{l}\sum_{t=1}^{l} \bar{h}_{a_t}, \mathbf{h_{qa}} = \frac{1}{n}\sum_{t=1}^{n} \bar{h}_{qa_t}, \mathbf{h_{aq}} = \frac{1}{l}\sum_{t=1}^{l} \bar{h}_{aq_t}, \tag{3.18}$$

Notice that for article sentence representation, we do not use the contextual sentence representation y_i from the hierarchical model because the idea is to exploit information of word-level representation $h_{i,j}^{enc}$ of sentences. The word-level representation contains more fine-grained information. The decoder of the model utilizes sentence-level representation. Finally, the gated mechanism is applied to different pairs of fixed vector representations:

$$q_F = Gtd\_Mchsm(\mathbf{q}, \mathbf{h_q}) \tag{3.19}$$

$$a_F = Gtd\_Mchsm(\mathbf{a}, \mathbf{h_a}) \tag{3.20}$$

$$qa_F = Gtd\_Mchsm(\mathbf{a}, \mathbf{h_{qa}}) \tag{3.21}$$

$$aq_F = Gtd\_Mchsm(\mathbf{a}, \mathbf{h_{aq}}) \tag{3.22}$$

Sentence Decoupling:

Inspired by how humans extract distractors (see section 3.1), we derive a function to score the sentences according to their fitness for generating distractors.

$$m_i = \lambda_q s_i^T W_z q_F - \lambda_a (s_i^T W_z a_F + s_i^T W_z qa_F + s_i^T W_z aq_F) + b_z \tag{3.23}$$

Here, λ_q and λ_a are hyperparameters. $W_z \in \mathbb{R}^{rxr}$ and $b_z \in \mathbb{R}^r$ are learnable parameters. Similar to the approach of [GBL+19], we applied temperature τ to find the final softsel matching score η. Intuitively, softsel matching scores should decouple passage sentences that have the correct answer and potential candidate sentences (in context with the question) for distractor generation. Moreover, if a question can be answered using a few sentences, then the distribution of scores for those sentences should be high. In other cases, if the question requires reasoning or a summary of the article, then distribution should be inclined toward uniformity.

$$\tau = \sigma(W_q^T h_q + b_q) \tag{3.24}$$

$$\eta_i = m_i / \tau \tag{3.25}$$

where σ is the sigmoid activation function, h_q is contextual representation from word-level LSTM, W_q and b_q are parameters.

3.4.3 Hierarchical Multi-Decoder

We now discuss the different stages of our decoder network. Unlike previous works where a single decoder and Jaccard similarity over beam samples were used to generate three different distractors, we propose a novel multi-decoder model to generate distractors. We used three separate decoders (i.e., uni-directional LSTM networks) to generate three different distractors. *Ideally, multiple decoders should not focus on the same word of a sentence to generate distractors, and at the same time, they should not consider non-relevant sentences of the article.* We achieve this goal in two ways: (a) During training, we learn the parameters of the second and third decoder in such a way that the generated distributions over candidate words should not be exactly similar to the distributions generated by the earlier decoders and neither be very different and (b) as the ground truth for each decoder output is the same, we proposed a dis-similarity based loss function (we will provide more detail in section 3.4.4) to learn appropriate distributions.

Question Context Initialization

To ensure that the decoders start with the context of the question, we use a separate uni-directional LSTM layer ($LSTM^{init}$) to encode the question and use the last hidden state of LSTM (as also done in [GBL+19]) in two ways. (a) For each decoder, we use the last token of question q_{last} and (b) employed final cell state c_n^{init} and hidden state h_n^{init} of LSTM to initialize each decoder.

We further examined *whether using more than one last token of the question improves the question context or not.* In our limited experiments, we found that in this setting, the generated distractors were biased towards the question. So, there should be a trade-off between the question context and the quality of the generated distractor. Experimental results gave evidence that using the last token of questions works well.

Multi-Decoder Model

For a given decoder, at every decoding time step t we obtain two attention scores, i.e., sentence-attention score $\beta_i^{d_k}$ and word-attention score $\alpha_{i,j}^{d_k}$ for article sentences and article words respectively. The superscript k indicates k^{th} decoder. These attention scores are further combined with softsel matching score η_i to obtain final attention distribution $\bar{\alpha}_{i,j}^{d_k}$ as we describe in Equation 3.23. Combining a softsel matching score is necessary because we need to de-emphasize the sentences and words that are not in the question context or are answer-revealing. The idea of sentence and word-level attention is well studied as a hierarchical attention model for text summarization tasks and is proven to be efficient [SBV+15a]. We utilized article sentence representation y_i to compute the sentence-attention score.

h_t^{d1}, h_t^{d2} and h_t^{d3} are states from three decoder LSTMs at any given time-step t. As discussed previously, the learned distribution of different decoders should not be too similar or too different; we achieve this goal in the following way. The sentence and word level attentions for the first decoder are computed as:

$$\beta_i^{d_1} = y_i^T W_{d_1} h_t^{d_1} \tag{3.26}$$

$$\alpha_{i,j}^{d_1} = h_{i,j}^{encT} W_{d'_1} h_t^{d_1} \tag{3.27}$$

The attention scores for the second decoder are given by:

$$\beta_i^{d_2} = y_i^T W_{d_2} h_t^{d_2} - \lambda_{dist1} * y_i^T W_{d_1} h_t^{d_1}, \tag{3.28}$$

$$\alpha_{i,j}^{d_2} = h_{i,j}^{encT} W_{d'_2} h_t^{d_2} - \lambda_{dist1} * h_{i,j}^{encT} W_{d'_1} h_t^{d_1} \tag{3.29}$$

The second terms in Equations 3.28 and 3.28 try to move the distributions away from the distributions learned by the first decoder. Similarly, the set of equations for the third decoder is given by:

$$\beta_i^{d_3} = y_i^T W_{d_3} h_t^{d_3} - \lambda_{dist1} * y_i^T W_{d_1} h_t^{d_1} - \lambda_{dist2} * y_i^T W_{d_2} h_t^{d_2}, \tag{3.30}$$

$$\alpha_{i,j}^{d_3} = h_{i,j}^{encT} W_{d'_3} h_t^{d_3} - \lambda_{dist1} * h_{i,j}^{encT} W_{d'_1} h_t^{d_1} - \lambda_{dist2} * h_{i,j}^{encT} W_{d'_2} h_t^{d_2} \tag{3.31}$$

Where W_{d_1}, $W_{d'_1}$, W_{d_2}, $W_{d'_2}$, W_{d_3} and $W_{d'_3}$ are learnable parameters. λ_{dist1} and λ_{dist2} are hyper-parameters. Finally, we combine the $\beta_i^{d_k}$ and $\alpha_{i,j}^{d_k}$ and softsel matching score η_i. We further normalize the score to obtain final word-level attention distribution $\bar{\alpha}_{i,j}^{d_k}$ across the article.

$$\bar{\alpha}_{i,j}^{d_k} = \frac{\alpha_{i,j}^{d_k} \beta_i^{d_k} \eta_i}{\sum_{i,j} \alpha_{i,j}^{d_k} \beta_i^{d_k} \eta_i} \tag{3.32}$$

Then, the context vector c_t^k is derived using attention-weighted additive operation over the word-level context representations of article words.

$$c_t^k = \sum_{i,j} \bar{\alpha}_{i,j}^{d_k} h_{i,j}^{enc} \tag{3.33}$$

The distribution over given vocabulary words V at a time step t and for k^{th} decoder is computed as:

$$Pv_t^k = softmax(W_v tanh(W_{\bar{h}}[h_t^{d_k}; c_t^k]) + b_v) \tag{3.34}$$

Where W_v, $W_{\bar{h}}$ and b_v are learnable parameters.

3.4.4 Training and Dis-Similarity Loss

As the ground truth distractor for each decoder is the same, we need to be cautious that all decoders do not try to generate the same/similar (ground truth) distractor. In an attempt to achieve this, the parameters of the model are already computed in such a way that the parameters (α, β values) of any decoder D_k are dependent on previous decoders $D_{1 \cdots k-1}$. Towards this, additionally, we give an incentive to the model to learn slightly different distributions from the ground truth distractor during the learning process. We add a dis-similarity loss in the loss function to achieve this. The dis-similarity loss measures the distance between the ground truth distractor and the generated distractor. Hence, the loss function has two components: (a) cross-entropy loss and (b) dis-similarity loss, which are contrasting in nature. The impact of the dis-similarity loss is tuned using a parameter λ_{ds}. The effect of the addition of the dis-similarity is analyzed with detailed evaluations in the Results Section.

Dis-similarity loss is obtained in the following way:

1. First, we feed the ground truth distractor through a uni-directional LSTM network where the last hidden step $h_{end}^{d_g}$ encodes the contextual representation of the ground truth distractor.

2. We also collect the last hidden state representation from each decoder, i.e., $h_{end}^{d_1}$, $h_{end}^{d_2}$ and $h_{end}^{d_3}$.

3. Finally, a cosine similarity score is computed across ground truth representation and the final state of each decoder.

$$ds_i = cos(h_{end}^{d_g}, h_{end}^{d_i}) \tag{3.35}$$

where ds_i indicate similarity score with i^{th} decoder.

The final loss function to be minimized is given by:

$$L = \sum_{k=1}^{3} \left(- \sum_{D_k \in V} logP(D_k|S, Q, A; \theta_k) - \lambda_{ds} * (1 - ds_k) \right) \tag{3.36}$$

3.5 Experimental Setting

3.5.1 Dataset

We used two distractor generation datasets to evaluate the performance of the proposed models: (1) RACE DG and (2) RACE++ DG. RACE [LXL+17] is a reading comprehension dataset collected from English examinations of the middle school (called RACE-M) and high school (called RACE-H) Chinese students. It consists of 97,687 questions from 27,933 articles. RACE++ is an extension of the RACE dataset. RACE++ additionally includes 14,122 questions from 4,275 articles collected from college-level English examinations (called RACE-C). In the RACE dataset, each record is a 6-tuple containing the article, question, correct answer, and three distractors. It was observed that many distractors do not have any semantic relevance with the article [GBL+19]. Gao et al. [GBL+19] used linguistic features and handcrafted rules to eliminate poor-quality distractors. Designing an exhaustive set of rules is not a trivial task, and created rules may be biased in nature. Hence, we investigate at the semantic level filter good-quality distractors. To find semantically relevant distractors, we applied the following methodology:

1. We manually removed those distractors that are dependent on other distractors. For example, distractors like "all of the above," "both option a and option b are correct," and so on.

2. Distractor, question, and answer should have a minimum word/token length of three.

3. We removed questions that have fill-in-the-blanks at the beginning or in the middle of the question. Questions with fill-in-the-blanks in the last are retained.

4. For an <article, question, correct answer, distractor> tuple, we found BERT representations for each of these components and also of that of the individual article sentences. We then computed the cosine similarity of the distractor with

Parameters	RACE DG	RACE++ DG
Total no. of train samples	96501	135321
Total no. of dev samples	12089	16915
Total no. of test samples	12284	16915
Avg. article length (tokens)	342.0	342.3
Avg. question length	9.76	10.9
Avg. answer length	8.63	8.00
Avg. distractor length	8.48	7.68
Avg. sentences length (in the article)	19.9	19.6
Avg. no. of distractors per triplet	2.1	2.3

Table 3.1: Statistics of RACE and RACE++ DG dataset. Average statistics are computed across all three splits.

the question, correct answer, and each sentence of the article. The distractor is retained only if its cosine similarity with the question, correct answer, and the average cosine similarity over the article sentences - all of them were above a certain threshold (approximately 90% similarity).

After pre-processing, the modified RACE data is called RACE-DG, and RACE++ is called RACE++-DG. We randomly divide RACE++ DG data into the train (80%), test (10%), and validation (10%) splits. For RACE DG data, these splits were publicly available; we have used those. Statistics of the datasets are presented in Table 3.1.

3.5.2 Baselines

We compared the proposed model performance with the following baselines:

- **Seq2Seq** [LPM15]: It is standard encoder-decoder model with global attention mechanism. It consists of LSTM in both the encoder and the decoder side.

- **HieRarchical Encoder-Decoder (HRED)** [SBV+15a]: This is an advancement over the basic seq2seq model with global attention to handle large input. By construction, it is hierarchical to encode the input at word-level and sentence-level.

- **Hierarchical Static Attention (HSA)** [GBL+19]**:** This is similar to HRED but uses static and dynamic attention instead of single global attention.

- **Hierarchical Co-Attention (HCA)** [ZLW20]**:** It is an improvement over the HSA model by exploiting rich interaction between article and question by co-attention model.

- **Static Attention + Multi-Decoder (SMD) :** This is a variant of the proposed HMD-Net model. It employs static attention (as used in HSA [GBL+19]) instead of a softsel and gated mechanism on the encoder side. We define this model to check the effectiveness of the multi-decoder model in the previous literature model.

- **Encoder of HMD-Net + Decoder of HSA (EHMD+DHSA)**: We propose this variant to verify the effectiveness of the encoder of HMD-Net. The encoder of the model is that of the HMD-Net (utilizing softsel and gated contextual ideas), and the decoder is similar to the HSA model (single decoder that generates three distractors using a beam search algorithm).

Additionally, HMD-Net+LF, EHMD+DHSA+BERT, and HMD-Net+BERT baseline models are also developed with linguistic features (LF) and BERT for comparison.

3.5.3 Evaluation Metrics

We evaluated the performance of all the models on *seven* automated and *three* manual evaluation metrics. Unlike previous approaches, which use only word-overlap-based automated evaluated metrics like BLEU(1-4) [PRWZ02a] and ROUGE-L [Lin04a], we additionally consider lexical similarity and embedding-based metrics. These word-overlap-based metrics may not reflect actual model performance. Several drawbacks of the BLEU scores have been discussed in the literature [CBOK06]. We aim to report the scores based on metrics that reflect the actual performance of the system and correlate with human judgments. To accomplish this, we additionally use lexical similarity-based metric METEOR [LD09], embedding-based metrics [LLS+16], and BERT cosine similarity (BERT CS) metric. Unlike BLEU, METEOR leverages linguistic resources like Word-Net and the root form of the word to compute the score. The three embedding-based metrics are Greedy Matching [RL12], Embedding Average [WBGL16], and Vector Extrema [FPLT14]. Finally, we computed the BERT-CS score, influenced by the recent work on the BERT model [DCLT18]. We obtained the sentence representation from BERT for both generated and reference distractors and applied cosine similarity to compute the BERT-CS score.

For manual evaluation, we used *Grammatical correctness* and *Distractability.* Additionally, we performed another human assessment to identify which method is generating a more confusing distractor. *The more confusing the distractors are, the better the model is.*

3.5.4 Implementation Details

We modified the implementation of the OpenNMT toolkit [KKD+17] for the model development. The vocabulary of the model is 50,000 most frequent words from the training corpus. For the BERT-based model, we extracted the word feature from `bert-base-uncased` of dimension 768. For other models `GloVe.840B.300d` pre-trained word embedding [PSM14] is used. The Out-of-Vocabulary tokens are labeled as special symbol UNK. The number of layers for all the word encoders (either BiLSTM or uni-directional LSTM), including question-context initializer and target sentence encoder, is 1 and 2 for sentences. All three decoders have the number of LSTM layers as 2. We set 700 hidden sizes for both BiLSTMs (350 for each direction) and uni-directional LSTMs. After several experiments on the validation set, the hyper-parameters λ_q, λ_a, λ_{dist1}, λ_{dist2} and λ_{ds} are set as 0.5, -0.3, 0.5, 0.4 and 0.0001 respectively. 0.3 is dropout probability, and the gradient norm upper bound is set to 5. Except for word embedding, all the trainable parameters are initialized with $\mathcal{U}$(-0.1, 0.1). The stochastic gradient descent (SGD) optimizer is initialized with a learning rate of 0.1 for all the models. Mini batch size is set to 16. We run the model for 200k steps. After 150k steps, the learning rate is halved at every 10k steps till the end. Additionally, we employed teacher-forcing. The maximum length of the generated distractor is set to 15. The beam size is set to 10. All the hyper-parameters are searched over the validation split, and results are reported on the test split.

3.6 Results And Analysis

3.6.1 Automatic Evaluation Results

The automatic evaluation results of our proposed models are reported in Table-3.2 and Table-3.3 on RACE-DG and RACE++DG datasets, respectively. The comparison with literature methods is presented for only RACE-DG datasets. Due to computational constraints and high overlap between these two datasets, the best-performing baselines with RACE-DG data are considered baselines with RACE++DG datasets. It can be observed that HMD-Net outperformed all baseline models across three distractors. Linguistic features (LF) and BERT contextual embedding inclusion further improve model performance. HMD-Net+BERT emerged as our best-performing model, whereas the EHMD+DHSA+BERT model was the second-best model. We can observe that there is a significant performance gap between HMD-Net+LF and HMD-Net+BERT, which reveals the importance of contextual embedding. The bet-

	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	Embd Avg	G. Match	Ext.Score	BERT-CS
1st	Seq2Seq [LPM15]	25.28	12.43	7.12	4.52	13.58	-	-	-	-	-
	HRED* [SBV+15a]	27.96	14.41	9.05	6.35	14.68	-	-	-	-	-
	HSA* [GBL+19]	28.18	14.57	9.19	6.43	14.89	-	-	-	-	-
	HCA [ZLW20]	28.65	15.15	9.77	7.01	15.39	-	-	-	-	-
	EHMD+DHSA	28.25	14.52	9.34	6.66	24.03	10.76	0.569 ± 0.00006	2.530 ± 0.0004	0.357 ± 0.00005	0.813
	SMD	28.78	15.60	10.12	7.26	25.59	11.22	0.574 ± 0.0006	2.585 ± 0.0004	0.362 ± 0.00005	0.817
	HMD-Net	29.26	16.16	10.16	**7.66**	25.78	11.58	0.582 ± 0.00006	2.619 ± 0.0004	0.367 ± 0.00005	0.818
	HMD-Net+ LF	29.80	16.31	10.64	7.57	26.31	11.56	0.581 ± 0.00006	2.629 ± 0.0004	0.367 ± 0.00005	0.823
	EHMD+DHSA+BERT	29.44	16.02	10.06	6.6	25.04	11.08	0.586 ± 0.00005	2.610 ± 0.0004	0.364 ± 0.00005	0.823
	HMD-Net+ BERT	**30.99**	**17.30**	**11.09**	7.52	**26.50**	**12.07**	**0.591 ± 0.00005**	**2.667 ± 0.0004**	**0.370 ± 0.00005**	**0.823**
2nd	Seq2Seq [LPM15]	25.13	12.02	6.56	3.93	13.20	-	-	-	-	-
	HRED* [SBV+15a]	27.85	13.39	7.89	5.22	14.48	-	-	-	-	-
	HSA* [GBL+19]	27.85	13.41	7.87	5.17	14.41	-	-	-	-	-
	HCA [ZLW20]	27.29	13.57	8.19	5.51	14.85	-	-	-	-	-
	EHMD+DHSA	27.41	13.47	7.96	5.27	22.75	10.41	0.563 ± 0.00006	2.455 ± 0.0004	0.352 ± 0.00005	0.812
	SMD	28.17	14.62	8.96	6.00	24.15	10.82	0.570 ± 0.00006	2.519 ± 0.0004	0.355± 0.00005	0.814
	HMD-Net	28.84	15.06	9.29	6.37	24.79	11.15	0.580± 0.00006	2.591 ± 0.0004	0.364± 0.00005	0.818
	HMD-Net + LF	29.19	15.33	9.34	6.23	24.90	11.27	0.583 ± 0.00006	2.595 ± 0.0004	0.366 ± 0.00005	0.820
	EHMD+DHSA+BERT	30.16	15.9	9.68	6.19	24.05	11.29	0.583 ± 0.00005	2.535 ± 0.0003	0.359 ± 0.00004	0.823
	HMD-Net + BERT	**30.93**	**16.89**	**10.64**	**7.10**	**25.76**	**11.96**	**0.595 ± 0.00005**	**2.646 ± 0.0004**	**0.368 ± 0.00005**	**0.826**
3rd	Seq2Seq [LPM15]	25.34	11.53	5.94	3.33	13.23	-	-	-	-	-
	HRED* [SBV+15a]	26.73	12.55	7.21	4.58	14.86		-	-	-	-
	HSA* [GBL+19]	26.93	12.62	7.25	4.59	14.72	-	-	-	-	-
	HCA [ZLW20]	26.64	12.67	7.42	4.88	15.08	-	-	-	-	-
	EHMD+DHSA	26.93	12.97	7.32	4.56	22.31	10.29	0.560 ± 0.00005	2.416 ± 0.0003	0.352 ± 0.00005	0.811
	SMD	27.50	13.69	7.90	5.01	23.38	10.39	0.562 ± 0.00006	2.463 ± 0.0004	0.350 ± 0.00005	0.813
	HMD-Net	27.64	13.98	8.22	5.33	23.42	10.53	0.572 ± 0.00006	2.526 ± 0.0004	0.356 ± 0.00005	0.815
	HMD-Net + LF	29.09	14.64	8.63	5.60	24.63	10.99	0.580 ± 0.00005	2.540 ± 0.0004	0.360 ± 0.00005	0.819
	EHMD+DHSA+BERT	29.62	15.47	9.52	6.18	23.93	11.27	0.585 ± 0.00005	2.513 ± 0.0003	0.359 ± 0.00004	0.823
	HMD-Net + BERT	**29.70**	**15.95**	**9.74**	**6.21**	**24.91**	**11.37**	**0.584 ± 0.00005**	**2.614 ± 0.0004**	**0.363 ± 0.00005**	**0.824**

Table 3.2: Automatic evaluation results on the RACE-DG dataset. For Seq2Seq and HCA, the results are taken from [GBL+19]. For HRED and HSA (rows with *), the results are taken from [ZLW20] as these numbers are better than the numbers reported in the original paper [GBL+19] for the same dataset. Symbol (-) indicates that results are not available.

	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	Embd Avg	G. Match	Ext.Score	BERT-CS
1st	EHMD+DHSA	32.68	18.62	12.41	8.96	31.23	12.3	0.5713 ± 0.00005	2.4006 ± 0.0003	0.3649 ± 0.00004	0.8395
	SMD	33.18	18.45	11.43	7.36	**32.48**	12.53	0.5854 ± 0.00005	2.62 ± 0.0003	0.3770 ± 0.00004	0.8424
	HMD-Net	33.37	18.61	11.64	7.66	32.29	12.49	0.5877 ± 0.00005	2.6689 ± 0.0003	0.3766 ± 0.00004	0.8419
	HMD-Net+ LF	33.45	18.81	11.87	7.93	32.18	12.54	0.5826 ± 0.00005	2.6641 ± 0.0003	0.3762 ± 0.00004	0.8429
	EHMD+DHSA+BERT	33.57	19.38	12.79	8.96	31.81	12.44	0.5848 ± 0.00004	2.6624 ± 0.0003	0.3710 ± 0.00004	0.8444
	HMD-Net+ BERT	**34.58**	**20.26**	**13.54**	**9.66**	32.24	**12.85**	**0.5939 ± 0.00005**	**2.6904 ± 0.0003**	**0.3782 ± 0.00004**	**0.8452**
2nd	EHMD+DHSA	31.46	16.5	10.1	6.65	28.69	11.58	0.5656 ± 0.00005	2.3023 ± 0.0003	0.3540 ± 0.00004	0.8371
	SMD	32.42	17.29	10.36	6.54	30.41	12.09	0.5774 ± 0.00005	2.5257 ± 0.0003	0.3684 ± 0.00004	0.8422
	HMD-Net	33.99	17.62	10.42	6.45	30.49	12.23	0.5839+/- 0.00005	2.5756 ± 0.0003	0.3709 ± 0.00004	0.8423
	HMD-Net + LF	33.26	18.03	10.79	6.81	**31.01**	12.37	0.5846 ± 0.00005	2.6099 ± 0.0003	0.3763 ± 0.00004	0.8426
	EHMD+DHSA+BERT	33.47	18.83	12.28	8.5	29.51	12.21	0.5838 ± 0.00004	2.6248 ± 0.0002	0.3642 ± 0.00004	0.8425
	HMD-Net + BERT	**34.01**	**19.53**	**12.83**	**9.02**	30.86	**12.51**	**0.5953 ± 0.00004**	**2.6554 ± 0.0003**	**0.3763 ± 0.00004**	**0.8430**
3rd	EHMD+DHSA	31.27	15.85	9.38	6.05	27.67	11.49	0.5698 ± 0.00005	2.2752 ± 0.0002	0.3533 ± 0.00004	0.8362
	SMD	31.73	16.39	9.42	5.72	29.85	11.73	0.5794 ± 0.00005	2.483 ± 0.0003	0.3682 ± 0.00004	0.8376
	HMD-Net	32.14	16.67	9.55	5.69	29.75	11.95	0.5864 ± 0.00004	2.5196 ± 0.0003	0.3683 ± 0.00004	0.8369
	HMD-Net + LF	31.89	16.89	9.85	6.07	29.75	11.95	0.5736 ± 0.00005	2.5819 ± 0.0003	0.3653 ± 0.00004	0.8380
	EHMD+DHSA+BERT	33.26	18.59	12.05	8.32	29.12	12.14	0.5817 ± 0.00004	2.5675 ± 0.0002	0.3635 ± 0.00004	0.8401
	HMD-Net + BERT	**33.29**	**18.84**	**12.28**	**8.52**	**29.87**	**12.17**	**0.5881 ± 0.00005**	**2.6214 ± 0.0002**	**0.3690 ± 0.00004**	**0.8400**

Table 3.3: Automatic evaluation results of different models on RACE++ DG dataset.

Models	Annot-set1	Annot-set2	Annot-set3
SMD	24	27	25
HMD-Net	25	26	27
HMD-Net + LF	34	30	33
HMD-Net + BERT	37	37	35

Table 3.4: Comparative study results of human evaluation.

ter score of SMD over the HCA model across all three distractors acknowledges that the generation of distractors is suitable and effective in the multi-decoder setting. A higher score of EHMD+DHSA over the HSA model across all three distractors validates the impact of the stronger encoder. All three distractors have similar observations, and the results for the second and third distractors also improved significantly as compared to previous approaches.

The results across METEOR, embedding-based metrics, and BERT-CS are consistent and similar to lexical overlap-based metrics. These metrics give additional evidence that HMD-Net+BERT consistently outperforms all other models. The HMD-Net model scores for embedding metrics are close to baseline models. Considering this, we further investigate and obtain statistically significant bounds for each metric, which validate the correctness/reliability of the reported scores. Additionally, the very high BERT-CS scores (>0.81) confirm that generated distractors are semantically very close to reference distractors. The evaluation scores for all the models on the RACE++DG dataset are higher as compared to the RACE-DG dataset. This improvement in performance can be attributed to (a) the size of RACE++DG is bigger, and (b) it contains quality distractors that help the models to learn better.

3.6.2 Human Evaluation Results

With human evaluation, we try to find answers to the following questions: (a) *Which of the models is performing the best?*, (b) *What is the quality of the generated text?* and (c) *Do these evaluation scores correlate with automated evaluation?* To answer these queries, we have performed two types of human assessments: *comparative study* and *quantitative study.*

1. **Comparative Study**: For this study, we employed 30 annotators (holding at least a master's degree in computer science and fluent in the English language). The annotators were distributed in three annotator sets, each of size 10. From the RACE-DG evaluation dataset, we randomly selected 120 questions from

40 articles (three questions from each article). To reduce bias and subjective evaluation, we gave 120 questions to each annotator set. Every annotator had to annotate 12 questions from 4 passages. To each annotator, along with the passage and question, we provided four different distractors (as options) from our four models: SMD, HMD-Net, HMD-Net+LF, and HMD-Net+BERT. We asked annotators to select the closest correct answer. It was mentioned to the annotators that some questions might not have the correct answer; in that case, they had to select the closest option. *We hypothesize that the distractors that are close to the correct answer (selected by annotators) are more confusing.* The more confusing the distractors are, the better the model is. We intentionally did not expose the correct answers to evaluators, so the evaluation should not be biased. Table 3.4 includes the results of this study. Each entry in the table indicates the number of times the distractor generated by the model (in row) is selected as the correct answer by the annotator set (in column). Comparing across all three annotator sets, it can be concluded that the HMD-Net+BERT model generated more confusing distractors. The second best model is HMD-Net+LF. The performance ordering of the models is similar to the one obtained in automatic evaluation results.

2. **Quantitative Study**: To have a quantitative idea of the quality of generated distractors, we asked each annotator to rate the generated distractors on a scale of 1-to-5 (1 is very poor and five is very good) on two manual evaluation metrics: (a) **Grammatical correctness**- *how grammatically correct are the distractors?* and (b) **Distractability**- *how confusing are the distractors?* As the Quantitative Study will provide absolute evaluation scores, we conducted this on a large dataset and six models. We randomly selected 350 questions from 117 passages. We employed two sets of annotators (holding at least a master's degree in computer science and fluent in the English language), each having seven annotators. Each set of annotators had to evaluate all 350 questions. Each annotator had to annotate 50 questions. Due to more data and more number of models in this task as compared to the comparative study, the workload on the annotators was more for this task. In this study, we randomly selected one distractor from EHMD+DHSA, SMD, HMD-Net, HMD-Net+LF, EHMD+DHSA+BERT, and HMD-Net+BERT models. The outcomes of the study can be found in Table-3.5. The grammatical correctness metric received a high score over distractibility, and HMD-Net+BERT was the best-performing model. The grammatical correctness scores correlate with automated scores.

	Models	Annot-set1	Annot-set2
Grammatical Correctness	EHMD+DHSA	4.007	3.298
	SMD	4.058	3.894
	HMD-Net	3.780	3.747
	HMD-Net+LF	4.061	3.988
	EHMD+DHSA+BERT	4.054	**4.071**
	HMD-Net+BERT	**4.155**	3.982
Distractability	EHMD+DHSA	2.431	2.557
	SMD	2.567	2.457
	HMD-Net	2.522	2.491
	HMD-Net+LF	2.680	2.560
	EHMD+DHSA+BERT	2.661	**2.752**
	HMD-Net+BERT	**2.752**	2.634

Table 3.5: Quantitative study results of human evaluations

Considering the fact that the current performance ceiling of humans on the RACE dataset is 95% [LXL+17] (for identification of correct answer given article, question, and four options), confusing humans is a challenging task. Considering these factors, we can conclude that our model performed decently on the distractability aspect.

3.6.3 Ablation Study and Inter-distractor Similarity Test

To verify the effect of each component of the proposed HMD-Net, we performed an ablation study. The results on the first distractor can be seen in Table-3.6. It is observed that the evidence encoding layer, dis-similarity loss, and gated contextual representation are key components of the model. The removal of these components resulted in lower performance. Using the last two tokens of the question sentence - diverts the model training and the model performed worst under this setting. The possible explanation can be - including a larger context may disturb distractor sentence structure and the model gets confused in learning critical patterns. The removal of contextual evidence representation (gated mechanism output), h_{aq} and h_{qa} have a minor impact on the model. Overall, the proposed configuration performs best.

It is expected that generated distractors should be semantically related to each other. If all the generated distractors are similar on a lexical level, then all the metrics used above will report a high score for all the distractors, but effectively, only one distractor is generated. This error is hard to catch until some careful analysis is performed. To nullify this kind of situation and provide evidence that our generated distractors are different at the lexical level, we performed an additional experiment

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR
Full HMD-Net Model	29.26	16.16	10.16	7.66	25.78	11.58
Without EEL (with CR)	28.60	15.17	9.68	6.90	25.15	10.96
Without CR (with EEL)	29.15	15.71	10.20	7.27	25.62	11.40
*Without CER	28.86	15.46	9.96	7.15	25.38	11.11
Without h_aq & h_qa	28.92	15.89	10.39	7.50	25.56	11.44
Without DSL	28.35	15.24	9.91	7.05	25.39	10.96
With last two Question tokens in QCI	20.90	9.38	5.45	3.50	17.45	8.24

Table 3.6: Ablation study results of the first distractor on the RACE DG dataset. Where EEL is the evidence Encoding Layer, CR is contextual representation, CER is contextual evidence representation (gated mechanism output), DSL dis-similarity loss, and QCI question Context Initialization. The row with * indicates that -the results are obtained without CER where the SS matching score is obtained from the average of CR and EEL scores

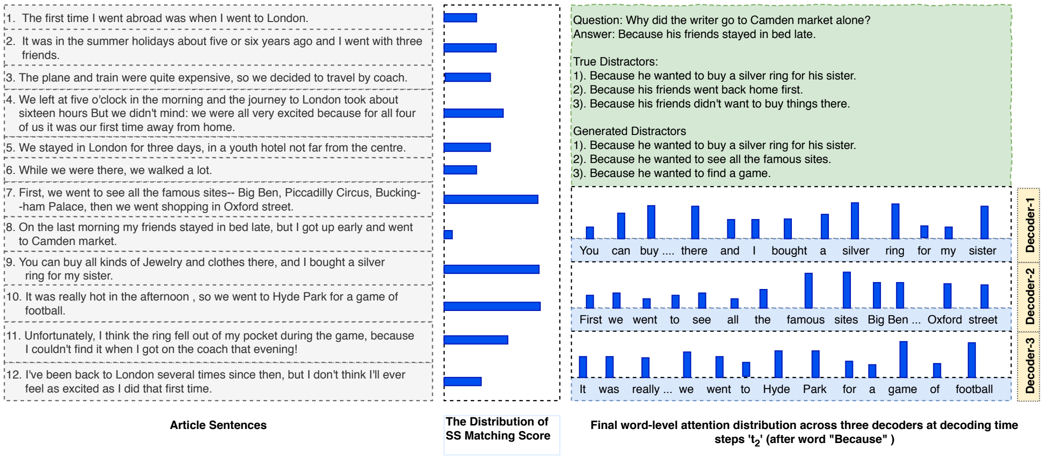
Models	**Dist1 & Dist2**	**Dist1 & Dist3**	**Dist2 & Dist3**
SMD	0.200	0.191	0.216
HMD-Net	0.221	0.210	0.236
HMD-Net + LF	0.215	0.219	0.201
HMD-Net + BERT	0.264	0.251	0.246

Table 3.7: Average Jaccard Similarity scores across generated three distractors on RACE-DG dataset.

on the RACE-DG dataset. For each pair of generated distractors, we computed the Jaccard Similarity (JS). The results are reported in Table-3.7. Note that the maximum similarity was 0.264 on the BERT model, which is very low. Previously, [GBL+19] and [ZLW20] used the JS threshold as 0.5 while selecting three distractors from the pool of distractors generated by the beam search algorithm. This gives evidence that our model generates lexically distinct distractors.

3.6.4 Case Study

Fig. 3.4 presents a sample distractor generated from the HMD-Net model. In the middle, we plotted the distribution of the softsel matching score (SSMS). We can observe that sentences 7, 9, and 10 are potential candidates for distractor generation and received high scores. Sentence 8 contains the correct answer and hence received a very low score. The three ground truth and generated distractors are shown on the top right side of the figure. In the bottom right, final learned attention ($\bar{\alpha}_{i,j}^{d_k}$) is included for all three decoders at decoding time step t_2, i.e., after generating word *'Because'*. For generating the next word, Decoder 1 selected Sentence 9, Decoder 2



1. The first time I went abroad was when I went to London.
2. It was in the summer holidays about five or six years ago and I went with three friends.
3. The plane and train were quite expensive, so we decided to travel by coach.
4. We left at five o'clock in the morning and the journey to London took about sixteen hours But we didn't mind: we were all very excited because for all four of us it was our first time away from home.
5. We stayed in London for three days, in a youth hotel not far from the centre.
6. While we were there, we walked a lot.
7. First, we went to see all the famous sites-- Big Ben, Piccadilly Circus, Bucking--ham Palace, then we went shopping in Oxford street.
8. On the last morning my friends stayed in bed late, but I got up early and went to Camden market.
9. You can buy all kinds of Jewelry and clothes there, and I bought a silver ring for my sister.
10. It was really hot in the afternoon , so we went to Hyde Park for a game of football.
11. Unfortunately, I think the ring fell out of my pocket during the game, because I couldn't find it when I got on the coach that evening!
12. I've been back to London several times since then, but I don't think I'll ever feel as excited as I did that first time.

Question: Why did the writer go to Camden market alone?
Answer: Because his friends stayed in bed late.

True Distractors:
1). Because he wanted to buy a silver ring for his sister.
2). Because his friends went back home first.
3). Because his friends didn't want to buy things there.

Generated Distractors
1). Because he wanted to buy a silver ring for his sister.
2). Because he wanted to see all the famous sites.
3). Because he wanted to find a game.

You can buy .... there and I bought a silver ring for my sister
Decoder-1
First we went to see all the famous sites Big Ben ... Oxford street
Decoder-2
It was really ... we went to Hyde Park for a game of football
Decoder-3
Article Sentences
The Distribution of SS Matching Score
Final word-level attention distribution across three decoders at decoding time steps 't2' (after word "Because")

Figure 3.4: An overview of distractor generation from HMD-Net model.

selected Sentence 7, and Decoder 3 selected Sentence 10. Each decoder obtains word distribution for the selected sentence and accordingly generates the next word.

3.7 Conclusion

In this paper, we have presented a Hierarchical Multi-Decoder Network (HMD-Net) and its variants with linguistic features and BERT contextual embedding. It is a data-driven approach to generate long and high-quality distractors from MCQ reading comprehension. We exploited the rich interactions among question, answer, and passage using the SoftSel operation and Gated Mechanism at the encoder side. On the decoder side, we used three separate decoders to generate three distractors. The distractors should not be lexically similar to each other or not very different. Our model outperformed all baselines across automated and manual evaluation metrics. We also prepared a high-quality new distractor generation dataset, RACE++DG.

3.8 Insights, Limitations and Future Work

Insights: One of the challenges we face in extracting contextual word embeddings from the BERT model is due to BERT's tokenization process, which breaks a word into subwords. Keeping track of word mapping to subword splits was challenging in the past, unlike today, where this feature is readily available in the Hugging Face library [Jai22]. We explicitly introduce a flag to identify the word-to-subword mapping and then perform average pooling of hidden representations across subwords to obtain the final contextual representation of the word from the BERT model. The other insight was the BERT-based model run-time was approximately 2 times that of traditional word embedding models, possibly due to the large embedding dimension.

Limitations and Future Work: One of the limitations of the proposed model is the extension of distractor generations to an arbitrary number of distractors. This requires additional decoders and re-training of the model, which can be costly. A hopeful direction is to add an additional diversifying module with a stranded encoder-decoder model. Diversifying modules can trigger diverse generations without additional decoders. We have made a similar effort in [EMKD23] for diverse headline generation that mitigates this limitation. Additionally, with the modern large language model, the representation of the input can be obtained easily, and overall modeling can be simplified. We have left this for future work.

Chapter 4

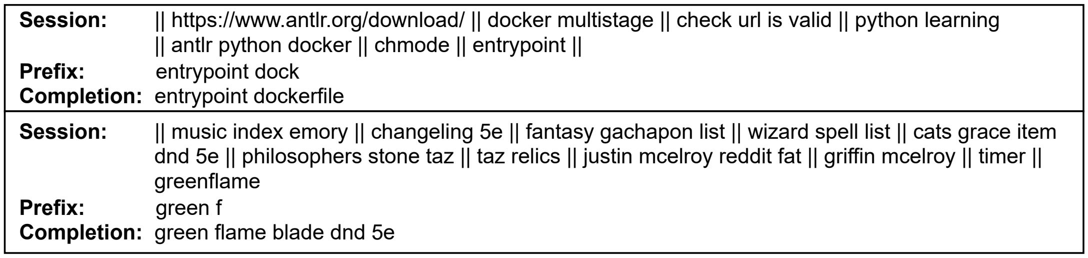
Advancing Frontiers of NLG: Personalized Query Auto-Completions

4.1 Introduction

In this chapter, we continue to advance natural language generation (NLG) modeling. We focus on the task of personalized Query Auto-Completion (QAC), which is a key functionality of modern search engines. Unlike the previous chapter, this is a more recent research effort that leverages large language models and trending modeling techniques. In particular, we explore the Retrieval-Augmented Generation (RAG; [KLJ+20, BMH+22]) modeling framework to improve the performance of personalized QAC systems. Before delving into modeling details, let us first understand what the PQAC task is and how the *limited context* in PQAC affects the system's performance.

Query formulation could be time-consuming for naïve users or users with complex information needs. Modern search engines, therefore, have a Query Auto-Completion (QAC) module to assist users in efficiently expressing their information needs as a search query. The goal is to help users finish their search task faster by accurately understanding their query intent using the partially typed prefix. While users type a partial search query (i.e., query prefix), the QAC system recommends a list of relevant and complete queries (i.e., query auto-completions or suggestions).

Most of the popular search engines adopt a two-stage approach for QAC: *candidate retrieval* and *candidate ranking* [CDR16]. A set of prefix-preserving suggestions is re-



Session:	|| https://www.antlr.org/download/ || docker multistage || check url is valid || python learning || antlr python docker || chmode || entrypoint ||
Prefix:	entrypoint dock
Completion:	entrypoint dockerfile
Session:	|| music index emory || changeling 5e || fantasy gachapon list || wizard spell list || cats grace item dnd 5e || philosophers stone taz || taz relics || justin mcelroy reddit fat || griffin mcelroy || timer || greenflame
Prefix:	green f
Completion:	green flame blade dnd 5e

Figure 4.1: Examples of session, prefix and completion. The session can have multiple previously user-typed queries. Here, we consider the session which has heterogeneous queries. Session queries are separated by '||'.

trieved from a pool of complete candidate queries in the candidate retrieval stage[1]. Typically, this is supported using a trie that records complete suggestions along with their historical popularity scores computed over a time window. Candidate retrieval could leverage various heuristics like historical candidate popularity, language or region-based affinity, freshness, etc. In the candidate ranking stage, these retrieved queries are ranked based on a larger list of features, including popularity, the user's previous search intent, the user's profile, etc. Finally, top-N-ranked candidates are shown to the user.

Although QAC has been studied for many decades, there are two major challenges yet to be solved.

1. **Short Prefixes:** High-quality completions for very short prefixes are the most desirable feature for any QAC system. But short prefixes are likely to have a huge candidate pool from the trie, and most of the QAC models return the most popular completions that may not be relevant.

2. **Unseen prefixes:** Trie-based systems fail to provide recommendations for prefixes that have never been recorded previously, i.e., not a part of the query log. We refer to such prefixes as *unseen* prefixes.

To overcome these problems, more recently, seq2seq neural models have gained attention [DRAF17, MLP20, YTZ^{+}20]. Besides the current prefix, these neural network-based NLG models are more powerful because they can also utilize relevant session information to recommend personalized query completions. But even NLG models have the following drawbacks: (1) Unlike trie-based methods, NLG models cannot directly incorporate historical popularity which is a very important signal. (2)

[1] A small percentage of suggestions are not prefix preserving; in this work, we focus on prefix-preserving suggestions only.

With increased levels of multi-tasking, sessions have become heterogeneous, diverse, and dynamic. This makes it difficult to focus on session queries relevant to the current prefix [YTZ+20]. A few such session examples are presented in Fig. 4.1. (3) Attention-based NLG methods that attempt to discover relevant session queries by computing similarity with prefix representations suffer when prefixes are too short. Misleading attention leads to poor completions and (4) As the unseen prefixes are typed rarely, corresponding session information may not be very relevant. In summary, the lack of proper context for short and unseen prefixes leads to poor performance in NLG-based PQAC systems.

NLG models are capable of capturing the semantic relationships between existing session queries, prefixes and completion. On the other hand, information in tries is like frequency-based high-confidence rules that capture relationships between prefixes and completions in a syntactic manner. *We hypothesize that jointly leveraging popularity signals from trie, semantic and personalization signals from previous session queries using an NLG mechanism is essential for effective QAC.* Based on this hypothesis, we propose a novel model for QAC, **Trie-NLG**, which uses a sequence-to-sequence Transformer architecture. To the best of our knowledge, such joint modeling of NLG techniques with popularity signals from trie for query auto-completion has not been studied in the literature.

We explore the RAG framework for the modeling. Given a prefix, Trie-NLG first extracts up to top-m most popular completions from the trie. We utilize a trie with around one billion suggestions constructed using 1.5 years of past query logs (Jul 2020 to Dec 2021) from Bing. For unseen prefixes, trie lookups lead to no (prefix-preserving) matches. To solve this problem, inspired by [MC15], we first index all suffix word n-grams from query logs into a suffix trie along with suffix popularity. We then look up unseen prefixes against the suffix trie to extract the top most popular synthetic completions. These m popularity-based completions, either from the main trie or from the suffix trie, are augmented as external context along with session queries and prefixes and passed as input to the seq2seq model. We hope that having *additional context* from trie-lookup will enable the NLG model to retain/copy good quality completions along with the generation of the novel but relevant completions.

Overall, our main contributions are as follows:

- We motivate the need for incorporating both popularity signals from tries and personalization signals from previous session queries for effective QAC, especially for short and unseen prefixes.

- We propose a novel architecture, Trie-NLG, which consists of a seq2seq Transformer model trained using rich context comprising of recent session queries and top trie completions. *To the best of our knowledge, this is the first attempt of trie knowledge augmentation in NLG models for personalized QAC.*
- Our proposed model provides state-of-the-art performance on two real prefix-to-query click behavior QAC datasets from Bing and AOL. We also perform several analyses including ablation studies to prove the robustness of the proposed model.

4.2 Related Work

In this section, we focus on three threads of related work for Query Auto-Completion (QAC), viz., traditional, learning-based, and language generation-based approaches.

4.2.1 Traditional Approaches for QAC

Most of the traditional QAC systems leverage *tries* [HO13] which store historical co-occurrence statistics of prefix and complete query pairs. The most popular QAC approach using trie lookups is *"Most Popular Completion"* (MPC; [BYK11]) which suggests top-N most popular (frequent) queries that start with the given prefix. Mitra et al. [MC15] extended this approach to generate candidates for rare prefixes using frequently observed query suffixes mined from historical search logs. On similar lines, other methods rely on term co-occurrence [HCO03], user click information [MZC08], clustering queries [SMWH10], and using word level representations [BPS$^+$12]. Some previous studies [BMM11, MBH17] also focused on modeling approaches when search logs are not available.

4.2.2 Learning-Based Approaches for QAC

Query log-based approaches are usually context-agnostic and suffer from data sparsity issues. It is critical to leverage context for capturing personalized intent and behavior. To cope with these limitations, different sources of knowledge have been exploited in the candidate ranking stage with the learning-to-rank framework [WBSG10]. These additional signals include session information [BYK11, JKCC14], user behavior [HMRS14, MSRH14], personalization [CLDR14, Sho13] and time/popularity-sensitivity [SR12].

Learning methods include LambdaMART [Bur10], logistic regression [Sho13], convolutional neural network (CNN; [MC15]), deep learning based ranking model (DRM; [ZZS+18]), and eXtreme Multi-Label Ranking [YSH+21]. These ranking models, however, fail to generate completions for unseen prefixes. Unlike these, we develop NLG models that capture personalization, learn contextual input representations, and provide completions even for unseen prefixes.

4.2.3 NLG-Based Approaches for QAC

Recently, sequence-to-sequence language model-based approaches have also been tried for QAC [PC17, WZM+18]. Given a prefix and optionally personalization information, these models generate prefix-preserving completions. These models can generate completions for unseen prefixes. Wang et al. [WZM+18] use LSTM (Long Short-Term Memory networks) and GRU (Gated Recurrent Units) based character-level language models to generate completions. Dehghani et al. [DRAF17] proposed GRUs with attention and copy mechanisms to incorporate the most prominent part of the previous queries. Mustar et al. [MLP20] and Yin et al. [YTZ+20] proposed Transformer [VSP+17] based models. Yin et al. [YTZ+20]'s approach requires additional browsed item information and also needs CTR values as labels to train the model. Moreover, these generation models still fail to generate good completions for short and rare prefixes. Unlike these methods, inspired by the RAG modeling framework [KLJ+20, BMH+22], we encode additional trie context along with a session in the NLG model, which leads to more meaningful completions for short, rare, and unseen query prefixes. Note that in our case, additional context is obtained from tries that are a part of any QAC system. This additional context mitigates the issue of limited context in QAC.

4.3 Problem Formulation

Consider a user u whose previous n queries (earliest to latest order) in the current session s are $\{q_1, q_2, ..., q_n\}$. The user is typing the current query q, where p is the query prefix typed so far. Additionally, there are *up to* m candidate query completions (top-ranked to low-ranked order) $\{c_1, c_2, ..., c_m\}$ available as additional context e from a trie. We aim to generate top-N query completions conditioned on current query prefix p, additional trie context e, and session information s. Mathematically, the task can be formulated as learning a model with parameters θ such that the probability of generating query q has to be maximized. The probability of generating query q is:

$$P_\theta(q|p; c_1, c_2, \ldots, c_m; q_1, q_2, \ldots, q_n) \tag{4.1}$$

Here, we consider the value of N to be equal to 8, i.e., the number of auto-completions is 8.

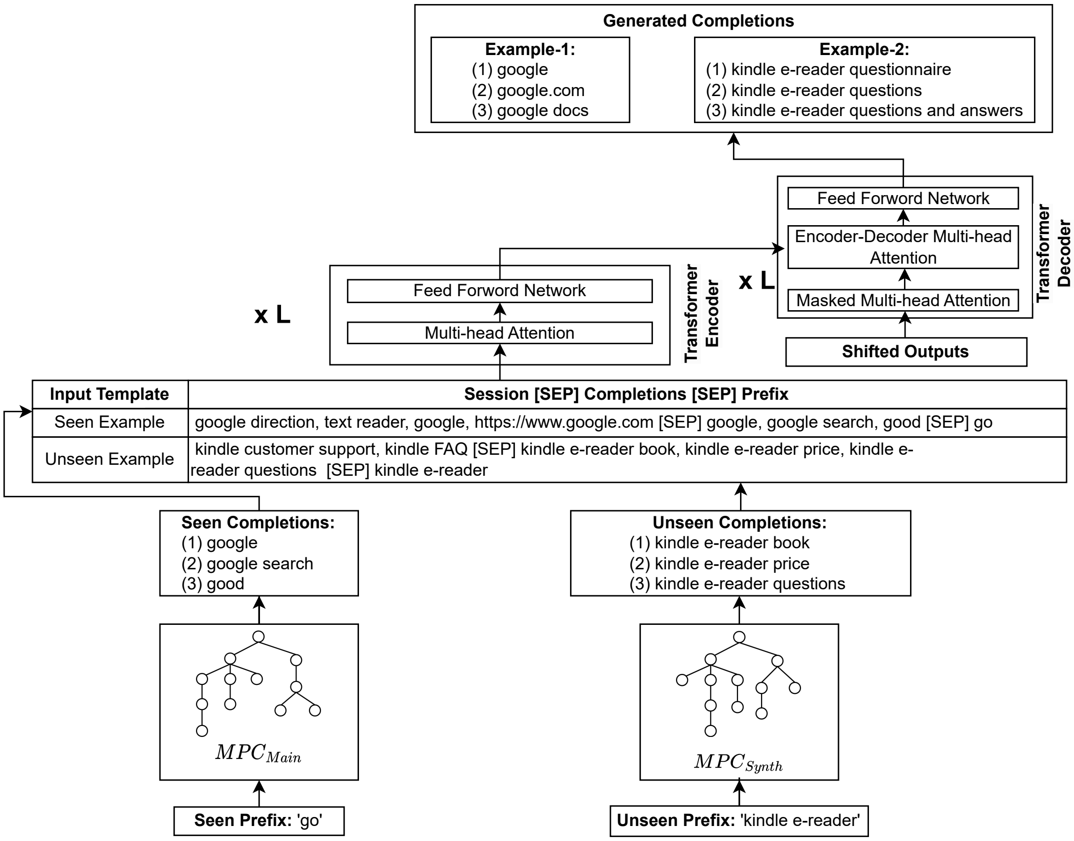
4.4 Methodology

The proposed TRIE-NLG model extracts a few completions from the trie and augments them as part of the input to an NLG model. For a given prefix, up to top-m completions are extracted as additional context from the trie using MPC. Those prefixes for which completions can be obtained from the MPC are called *Seen* prefixes, while those for which completions are not present are called *Unseen* prefixes. For seen prefixes, we leverage the main trie, and for unseen prefixes, we leverage a new trie called the suffix trie. These suggestions from the main or suffix trie are augmented with previous queries in the session and the current prefix and passed as input to the NLG model to generate accurate completions. Fig. 4.2 illustrates the overview of the proposed model. To enable a concrete understanding of the proposed model, we consider two running examples. For simplicity, we only consider the prefix and ground truth completion in the ⟨*prefix, completion*⟩ template. Example-1: ⟨*go, google.com*⟩ and Example-2: ⟨*kindle e-reader, kindle e-reader questionnaire*⟩.

4.4.1 Trie Context Extraction (MPC$_{\text{Main}}$)

To extend the context associated with seen prefixes, top-ranked completions are extracted from the main trie (called MPC$_{\text{Main}}$) which has been created using 1.5 years' worth of Bing query logs. Given a prefix p, MPC$_{\text{Main}}$ provides up to m completions $\{c_1, c_2,$ ..., c_m $\}$. In case the prefix is not present in the trie, the lookup will return no responses. For our running example-1, for the prefix *go*, MPC$_{\text{Main}}$ returns three completions: *google*, *google.com*, and *good*. However, for running example-2 prefix *kindle e-reader*, no completions are obtained from MPC$_{\text{Main}}$. The prefix *go* is referred to as *seen* prefix, while *kindle e-reader* is referred as *unseen* prefix.

$$\{c_1, c_2, \ldots, c_m\} = \text{MPC}_{\text{Main}}(p) \tag{4.2}$$



Input Template	**Session [SEP] Completions [SEP] Prefix**
Seen Example	google direction, text reader, google, https://www.google.com [SEP] google, google search, good [SEP] go
Unseen Example	kindle customer support, kindle FAQ [SEP] kindle e-reader book, kindle e-reader price, kindle e-reader questions [SEP] kindle e-reader

Figure 4.2: An overview of the proposed TRIE-NLG model

4.4.2 Synthetic Context Extraction ($\mathrm{MPC}_{\mathrm{Synth}}$)

For unseen prefixes, the $\mathrm{MPC}_{\mathrm{Main}}$ fails to provide any completions. In such cases, we make use of another trie called the suffix trie which is created by indexing all suffix word n-grams from query logs along with suffix popularity. Here, we consider all the queries across all the sessions from the training dataset. Since the suffix word n-grams may not be actual queries, we call them synthetic completions. Formally, if a query contains n words $\{w_1, w_2...w_n\}$, its substrings from each $i \in \{2, \cdots, n-1\}$ to n is called suffix. These suffixes are organized in another trie called the suffix trie (or $\mathrm{MPC}_{\mathrm{Synth}}$). For a given unseen prefix p_u, we look up the suffix $\mathrm{MPC}_{\mathrm{Synth}}$ to obtain the synthetic completions as:

$$\{c_1, c_2, \ldots, c_m\} = \mathrm{MPC}_{\mathrm{Synth}}(p_u) \tag{4.3}$$

For example, given a query *university of west florida*, the suffix trie will store synthetic completions like *florida*, *west florida* and *of west florida*. The suffix trie has key and associated values and frequency. For the above example, the suffix trie looks like this (simplified): *{ university: of west florida (1), university of: west florida (1), university of west: florida (1) }*. Frequency is indicated in the bracket. We lookup unseen prefixes against the suffix trie to extract top-m most popular synthetic yet useful completions. Note that these lookups still attempt to match the unseen prefix with prefixes of suffixes indexed in the suffix trie. In this way, we will be able to obtain completions for unseen prefixes that can not be obtained from MPC_{Main}. Although this idea is similar to one described by [MC15], unlike them, we consider the whole prefix and not only *end-term* of the prefix. If a prefix has multiple words, the last partial word is the end-term. The whole prefix has more meaningful contextual representation than end-term representation which leads to more accurate completions. For running example-2, $\text{MPC}_{\text{Synth}}$ returns three completions for the unseen prefix *kindle e-reader*: *kindle e-reader book*, *kindle e-reader price*, and *kindle e-reader questions.*

4.4.3 Context Augmentations in NLG

After obtaining trie suggestions, each data point consists of the session information (s), additional trie context (e), prefix (p), and the corresponding completion (q). We consider an Encoder-Decoder-based NLG model that takes the triplet $\langle s, e, p \rangle$ as input and attempts to generate the complete query (q). The input is provided to the model as a text sequence, where each element of the triplet is separated by a special token [SEP]. Trie context, i.e., top-m candidate completions are obtained from MPC_{Main} or $\text{MPC}_{\text{Synth}}$. During model training, the input triplet is first fed through the encoder to obtain a contextual representation. These contextual representations are semantic encodings of $\langle s, e, p \rangle$, which is key for the model's performance, particularly for short and unseen prefixes. Then, this contextual representation is passed through the decoder to generate top-N completions.

Relevant contextual suggestions from tries help the model with additional input that can guide the generation process. As typically the queries in a user session are often correlated in terms of the user's information need, the session context helps the model in understanding the user's current requirement. On the other hand, through the suggestions from the trie which is backed by historical query logs, a global perspective of the prefix and its possible completions preferred by a large user base can be obtained. The model thereby gets to see a local (concerning the user) as well as a global (concerning a large user pool) perspective surrounding the current prefix, and can appropriately utilize these inputs through language models pre-trained on large general-purpose corpora that understand semantic and syntactic aspects of natural language text. We hypothesize that the combination of these input and modeling choices makes the model superior for the target personalized QAC task.

The model is trained to maximize the probability of ground truth token sequence with maximum likelihood estimation (MLE). So, the following loss function is minimized:

$$L = -\sum_{i=1}^{D} \sum_{t=1}^{|y^i|} logP_t^i(\hat{y}_t^i | \hat{y}_{0:t-1}^i; p; e; s) \tag{4.4}$$

where D is the training dataset size, $|y^i|$ is the length of the i-th ground-truth query, $\hat{y}_t$ denotes token generated at time step t. P_t^i is the prediction probability distribution at t-th decoding step to generate the next token conditioned on previously generated tokens, prefix, trie context and session. The top-N completions are generated using beam search. For both running examples, both MPC_{Main} and $\text{MPC}_{\text{Synth}}$ failed to produce the correct ground truth completion. However, with context augmentation in NLG, it can be observed that for the prefix *go*, the final completion includes the correct completion *google.com.* Similarly, the prefix *kindle e-reader* has the correct completion *kindle e-reader questionnaire.* This demonstrates the effectiveness of augmenting additional context in the NLG for QAC systems.

4.5 Datasets and Experimental Setup

In this section, first, we will provide a detailed overview of the dataset, along with analyses. Then, we will provide details of the experimental setup, including the baseline, evaluation metrics, and other relevant information.

4.5.1 Datasets and Analysis

In this subsection, we present the details of the datasets, including data construction steps, pre-processing and some critical observations. We use two datasets: (1) Bing query log and (2) AOL public query log [PCT06]. The Bing dataset covers 9.08 million users, while the AOL dataset corresponds to 0.50 million users. The raw AOL query log consists of a sequence of queries entered by the users along with time-stamp details. We first pre-process the dataset by lower-casing all the queries, removing duplicate and single-character queries, and removing queries with a dominating (>50%) number of non-alphanumerics. Following previous studies [SBV+15b, YSH+21], we split the

Char	**Train**			**Validation**			**Test**		
Length	**Total**	**Seen**	**Unseen**	**Total**	**Seen**	**Unseen**	**Total**	**Seen**	**Unseen**
Total	20.40M	17.86M	2.54M	100K	92.43K	7.57K	100K	92.80K	7.20K
[1-5]	9.10M	8.80M	0.30M	40.68K	40.39K	0.29K	40.46K	40.19K	0.27K
[6-10]	4.30M	4.10M	0.20M	21.40K	21.07K	0.33K	21.62K	21.24K	0.38K
10+	7.00M	4.96M	2.04M	37.92K	30.97K	6.95K	37.92K	31.37K	6.55K

Table 4.1: Prefix distribution statistics for Bing dataset with prefix character length. 'M' and 'K' indicate that the value is in the order of millions and thousands respectively.

Char	**Train**			**Validation**			**Test**		
Length	**Total**	**Seen**	**Unseen**	**Total**	**Seen**	**Unseen**	**Total**	**Seen**	**Unseen**
Total	3.91M	3.47M	0.44M	100K	88.73K	11.27K	100K	88.69K	11.31K
[1-5]	1.42M	1.42M	0.00M	35.55K	35.54K	0.01K	36.59K	36.56K	0.03K
[6-10]	1.15M	1.11M	0.04M	29.53K	28.67K	0.86K	29.51K	28.61K	0.90K
10+	1.34M	0.94M	0.40M	34.92K	24.52K	10.40K	33.90K	23.52K	10.38K

Table 4.2: Prefix distribution statistics for AOL dataset with prefix character length. 'M' and 'K' indicate that the value is in the order of millions and thousands respectively.

sequence of queries into sessions with at least 30 minutes of idle time between two consecutive queries. We only retain those sessions which have at least two queries. Next, for a given session with queries (in earliest to latest order) $\{q_1, q_2, ..., q_n, q_{n+1}\}$ we create a triplet $r = \{(q_1, q_2, \ldots, q_n), p_{n+1}, q_{n+1}\}$ where p_{n+1} is a sampled prefix of query q_{n+1}. Sampling follows an exponential distribution favoring shorter prefixes. Each such triplet is a data point for modeling where input is all session queries except the last one, additional trie context, and prefix p_{n+1}. Ground-truth output is the last session query q_{n+1}.

Unlike the AOL dataset, where the prefix-to-query information is not explicitly available, and the prefixes are synthetically created by splitting a full query, the Bing dataset consists of real prefixes. Each example of the dataset consists of the user's session information s, current real prefix p_{n+1}, and real clicked completed query q_{n+1}. The dataset is obtained by considering only those cases where there was at least one past query in the user session, and the user has set their primary language as English.

Each dataset has two splits: *Seen Dataset* and *Unseen Dataset*. To obtain these splits, we use a trie (i.e., MPC_{Main}) with around one billion suggestions constructed using 1.5 years of past Bing query logs (Jul 2020 to Dec 2021). For a given prefix, if the trie contains at least one completion, then the prefix is called a *Seen Prefix*. Else, it is called an *Unseen Prefix*. The set of all Seen Prefixes (along with other search log attributes) is referred to as *Seen Dataset* and the set of all Unseen Prefixes is called *Unseen Dataset*. *Statistics of Bing and AOL datasets shown in Tables 4.1 and 4.2 respectively, indicate that there are 12.4% and 11.4% unseen prefixes in the training part of Bing and AOL datasets respectively.* The AOL and BING datasets each contain three sub-splits, namely train, validation, and test, which are based on temporal information and are applicable to both seen and unseen datasets as well. The timestamps for the train, validation, and test datasets are denoted as t_{train}, $t_{validation}$, and t_{test}, respectively. It is stipulated that t_{train} is the least recent of the three timestamps, and $t_{validation}$ and t_{test} are of increasing temporal proximity, with t_{test} being the most recent of the three.

To analyze the accuracy of various methods for different splits of the dataset, we created three prefix-length buckets: between 1 to 5 characters, 6 to 10 characters, and greater than 10 characters. Prefix distribution statistics with respect to each bucket are also reported in Tables 4.1 and 4.2 for Bing and AOL datasets, respectively. *We observe that ∼45% and ∼36% of the prefixes from the training datasets have lengths less than 6 for Bing and AOL, respectively.* This indicates the dominance of short prefixes and necessitates the design of better modeling techniques. An approach that provides additional context (as we proposed) is promising. We also observe that ∼80% and ∼91% of the unseen train dataset have prefix lengths 10+ characters for Bing and AOL respectively, which is another reason for them being less popular. The addition of more relevant context from tries may lead to better completions in such scenarios. We also observe that the average number of queries in a session for Bing is 5.2 and for AOL is 2.4. This provides diverse personalization contexts better to judge the applicability and usefulness of the models. Overall, unseen and short prefixes in QAC are frequent and challenging problems.

4.5.2 Evaluation Metrics

We evaluate all the baselines and proposed model with three evaluation metrics. To cover multiple aspects of the evaluation, we use both ranking-oriented metrics (MRR) and metrics to identify the quality of the generated sequence (BLEU and BLEU_{RR}).

1. **Bilingual Evaluation Understudy (BLEU):** It is a popular metric used for multiple NLG tasks. For our experiments, BLEU evaluates the degree of lexical

match between the ground-truth complete query and the first ranked generated query.

2. **Mean Reciprocal Rank (MRR):** MRR is one of the most popular metrics for evaluating ranking systems. MRR score is calculated as

$$\text{MRR} = \frac{1}{D_{ts}} \sum_{i=1}^{D_{ts}} \frac{1}{r_i} \tag{4.5}$$

   Here, D_{ts} is the size of the test dataset and r_i is the rank of the ground-truth complete query in the generated rank list for the i^{th} input. If the ground-truth complete query is not in the generated rank list, then r_i is set to ∞.

3. **BLEU Reciprocal Rank ($\mathbf{BLEU_{RR}}$; [YSH+21]):** It is defined as the reciprocal rank weighted average of BLEU score between the ground-truth query and generated completions.

$$\text{BLEU}_{RR} = \frac{1}{D_{ts}} \sum_{i=1}^{D_{ts}} \frac{\sum_{j=1}^{N} \frac{1}{j} \text{BLEU}(q, q'_{i,j})}{\sum_{j=1}^{N} \frac{1}{j}} \tag{4.6}$$

   where q is the ground-truth complete query and $q'_{i,j}$ is the j-th generated completion for the i-th test example.

4.5.3 Baselines

Our proposed model is based on both NLG and Trie models. Such joint modeling of NLG systems with popularity signals from trie has not been previously explored. To thoroughly evaluate its performance, we have carefully selected ten diverse baselines, including the traditional trie-based models (MPC, MPC+SynthMPC), ranking model (GRM), deep learning models (LSTM, Transformers) and pre-trained NLG models (T5, BART). In light of the superior performance demonstrated by transformer-based models, we have also included multiple strong transformer-based baselines. As our focus is on generation rather than ranking, we have selected more generative baselines for comparison. However, the outputs of our proposed model can be used as features in learning-to-rank and traditional models. The following baselines have been considered for comparison with the proposed model:

1. **MPC$_{\textbf{Train}}$:** This uses the traditional MPC method [BYK11]. The candidate rankings are obtained based on the popularity of each query from the historical query log. Here the historical query log is the training data itself.

2. **MPC$_{\textbf{Main}}$:** In this baseline the completions are obtained using the main trie created using 1.5 years of historical query logs from Bing.

3. **MPC$_{\textbf{Train}}$ + MPC$_{\textbf{Synth}}$/ MPC$_{\textbf{Main}}$ + MPC$_{\textbf{Synth}}$**: Completions are obtained using MPC$_{\text{Train}}$ and MPC$_{\text{Synth}}$/MPC$_{\text{Main}}$ for seen and unseen prefixes resp.

4. **GRM:** We first represent a session, prefixes and complete query as a bag-of-word (BOW) vector and then LambdaMART is trained with these features.

5. **Seq2Seq LSTM:** Standard LSTM based sequence-to-sequence model with attention. Input is a prefix and the target is the complete query.

6. **Seq2Seq Transformer:** Standard Seq2Seq Transformer model with architecture similar to T5-base. We train the model from scratch. Input is "session [SEP] prefix" and the target is the complete query.

7. **T5:** Same as Seq2Seq Transformer, except that we *fine-tune* T5-base [RSR+20b] on the QAC dataset.

8. **BART:** Similar to [MLP20], we fine-tune BART-base [LLG+19] with QAC dataset. Input and output are the same as that of T5.

9. **BART + Implicit Trie Context (ITC):** In this modeling, we try to augment trie's knowledge implicitly. It is a two-step training procedure: (i) BART-base is fine-tuned using a dataset, which consists of session and prefix as input and top m suggestions from MPC$_{\text{Main}}$/MPC$_{\text{Synth}}$ as the target. (ii) This training checkpoint is further trained with the QAC dataset, where the input is the session and the prefix and output are the clicked query. In the second step, we freeze the parameters of the first six decoder layers to retain the trie-based knowledge. During inferencing, the model checkpoint obtained from the second stage of training takes prefix and session as input and outputs the query suggestion.

10. **BART + MPC$_{\textbf{Main}}$:** This baseline augments the trie knowledge explicitly. With each training example, we add up to top-m trie completions as additional context. There are no completions for unseen prefixes. Input is session, prefix, and additional trie context and output is the clicked query.

4.5.4 Implementation Details

The proposed model and all the baselines are implemented in Python. GRM is implemented using learning to rank library[2] and seq2seq LSTM is implemented using Texar[3]. All the transformer-based models are implemented using HuggingFace Library[4]. All the experiments were conducted on eight A100 Azure cloud GPUs. The batch size is 128; the learning rate is 1e-4, the scheduler is 'linear,' the number of epochs is 5, and early stopping was enabled. We used the Adam optimizer with a max source length of 200 and a max target length of 32. BART-base has 6 layers, and 12 heads, layer normalization was enabled and the hidden layer dimension is 768. For the generation, the number of beam size (i.e., k) is 8, the maximum sequence length is set to 16, and the repetition penalty[5] is 0.6. We applied grid search for hyper-parameter tuning on the validation dataset and all the scores are reported on the test dataset. We experimented with 1, 3, 5, and 8 as values of m, the number of suggestions to extract from the trie. Based on the results of the validation dataset, $m = 3$ was selected for running experiments on the test data. We make our code publicly available[6].

4.6 Results and Discussions

In this section, we present and analyze results of different baselines and the proposed model.

4.6.1 Overall Performance Comparison

Tables 4.3 and 4.5 summarize the experimental results on the Bing and AOL datasets, respectively. Due to the confidential nature of the Bing dataset, we cannot report the exact values of the metrics. This is common practice in many previous studies [RJG+18] as well. Hence, in Table 4.3 and the rest of the paper we report percentage improvement scores of the models over reference $MPC_{Train} + MPC_{Synth}$ baseline for the Bing dataset. We cannot use $\text{MPC}_{\text{Train}}$ as a reference for showing percentage improvements as the model does not have any completions for unseen prefixes. For the publicly available AOL dataset, we report exact evaluation scores across all three

[2] https://github.com/jma127/pyltr

[3] https://github.com/asyml/texar

[4] https://huggingface.co/

[5] https://huggingface.co/blog/how-to-generate#appendix

[6] https://github.com/kaushal0494/Trie-NLG

	Seen + Unseen Dataset			**Seen Dataset**			**Unseen Dataset**		
Models	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**
MPC$_{\text{Train}}$	-7.90	-19.87	-24.64	0.00	0.00	0.00	-	-	-
MPC$_{\text{Main}}$	-0.11	36.61	26.38	8.49	70.38	63.68	-	-	-
MPC$_{\text{Main}}$ + MPC$_{\text{Synth}}$	7.78	56.42	47.01	8.49	70.38	63.68	0.00	0.00	0.00
GRM	-1.43	-5.32	-5.51	4.30	19.02	16.58	-	-	-
Seq2Seq LSTM	9.61	39.30	56.78	10.24	44.02	72.20	5.34	16.61	91.45
Seq2Seq Transformer	15.74	54.22	76.16	18.24	55.07	82.46	10.53	24.20	99.29
T5	20.78	61.81	78.70	20.76	70.83	85.13	21.60	27.33	103.98
BART	36.73	73.47	91.09	36.95	84.51	100.47	34.53	29.03	110.35
BART + ITC	31.66	71.43	88.72	31.58	81.97	96.84	33.05	29.12	111.00
BART + MPC$_{\text{Main}}$	54.12	86.77	110.78	56.14	101.35	129.00	34.10	28.04	110.80
TRIE-NLG	**56.78**	**88.26**	**114.52**	**56.56**	**101.99**	**130.04**	**59.74**	**33.02**	**123.07**

Table 4.3: Results of the models on Bing dataset. The reported scores are percentage (%) improvements over MPC$_{\text{Train}}$ + MPC$_{\text{Synth}}$ baseline. '-' indicates no completions are retrieved/generated for the model. Here we consider up to 3 completions as additional context from MPC$_{\text{Main}}$ or MPC$_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built.

	Seen + Unseen			**Seen Dataset**			**Unseen Dataset**		
Models	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\mathbf{RR}}$**	**ΔBLEU**
MPC$_{\text{Train}}$	-34.18	-55.37	-61.97	0.00	0.00	0.00	-	-	-
MPC$_{\text{Main}}$	-37.06	-18.18	-39.05	-4.31	83.60	52.64	-	-	-
MPC$_{\text{Main}}$+MPC$_{\text{Synth}}$	-2.87	37.19	19.10	-4.31	83.60	52.64	**0.00**	**0.00**	**0.00**
GRM	-30.35	-39.66	-48.40	6.03	36.06	19.70	-	-	-
Seq2Seq LSTM	40.25	21.48	28.25	87.93	111.47	152.23	-50.52	-50.08	-27.68
Seq2Seq Transformer	45.04	38.84	43.39	91.81	142.62	165.72	-45.48	-44.97	-23.03
T5	53.67	43.80	48.70	100.86	149.18	174.08	-37.5	-41.05	-19.31
BART	65.81	51.23	54.33	116.81	163.93	185.47	-32.03	-39.01	-16.84
BART+ITC	61.98	51.23	53.49	111.63	163.93	183.68	-33.29	-38.84	-17.16
BART+MPC$_{\text{Main}}$	69.96	53.71	55.81	123.27	168.85	**190.30**	-32.66	-39.35	-17.19
TRIE-NLG	**80.51**	**59.50**	**66.15**	**124.56**	**170.49**	**190.20**	-3.25	-29.81	-1.08

Table 4.4: Results of the models on AOL dataset. The reported scores are percentage (%) improvements over MPC$_{\text{Train}}$ + MPC$_{\text{Synth}}$ baseline. '-' indicates no completions are retrieved/generated for the model. Here we consider up to 3 completions as additional context from MPC$_{\text{Main}}$ or MPC$_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built.

	Seen + Unseen Dataset			**Seen Dataset**			**Unseen Dataset**		
Models	**MRR**	**$\mathbf{BLEU_{RR}}$**	**BLEU**	**MRR**	**$\mathbf{BLEU_{RR}}$**	**BLEU**	**MRR**	**$\mathbf{BLEU_{RR}}$**	**BLEU**
$\text{MPC}_{\text{Train}}$	20.6	5.4	15.25	23.2	6.1	19.49	-	-	-
$\text{MPC}_{\text{Train}}$ + $\text{MPC}_{\text{Synth}}$	31.3	12.0	40.10	23.2	6.1	19.49	**95.2**	**58.7**	**95.66**
MPC_{Main}	19.7	9.9	24.44	22.2	11.2	29.75	-	-	-
MPC_{Main} + $\text{MPC}_{\text{Synth}}$	30.4	16.6	47.76	22.2	11.2	29.75	**95.2**	**58.7**	**95.66**
GRM	21.8	7.3	20.69	24.6	8.3	23.33	-	-	-
Seq2Seq LSTM	43.9	14.7	51.43	43.6	12.9	49.16	47.1	29.3	69.18
Seq2Seq Transformer	45.4	16.8	57.50	44.5	14.8	51.79	51.9	32.3	73.62
T5	48.1	17.4	59.63	46.6	15.2	53.42	59.5	3461	77.18
BART	51.9	18.3	61.89	50.3	16.1	55.64	64.7	35.8	79.55
BART + ITC	50.7	18.3	61.55	49.1	16.0	55.29	63.5	35.9	79.24
BART + MPC_{Main}	53.2	18.6	62.48	51.8	16.4	**56.58**	64.1	35.6	79.21
Trie-NLG	**56.5**	**19.3**	**66.63**	**52.0**	**16.5**	**56.56**	92.1	41.2	94.62

Table 4.5: Results of the models on AOL dataset. Here we report exact evaluation scores, unlike the ones for the Bing dataset. We consider up to 3 completions as additional context from MPC_{Main} or $\text{MPC}_{\text{Synth}}$. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built.

metrics. We also report percentage improvement scores of the models over reference MPC_{Train} + MPC_{Synth} baseline for the AOL dataset in Table 4.4. Overall, our proposed Trie-NLG outperforms all the traditional, ranking, and generative models, across both the datasets (including Seen and Unseen) and all three metrics. Paired t-test shows that Trie-NLG outperforms the best baseline statistically significantly across both the datasets for each of the three metrics with a p-value less than 0.05.

Note that $MPC_{\text{Train}}+MPC_{\text{Synth}}$ and $MPC_{\text{Main}}+MPC_{\text{Synth}}$ have identical results for "unseen" datasets. Similarly, MPC_{Train} and $MPC_{\text{Train}} + MPC_{\text{Synth}}$ yield same results for "seen" dataset. This is expected because MPC_{Train} and MPC_{Main} provide trie suggestions for seen prefixes; while MPC_{Synth} provides trie suggestions for unseen prefixes.

Without $\text{MPC}_{\text{Synth}}$, the $\text{MPC}_{\text{Train}}$ and MPC_{Main} do not have completions for the Unseen dataset. $\text{MPC}_{\text{Synth}}$ provides completions for the unseen prefixes and boosts the overall model performance. As expected, the generative models provide suggestions for unseen prefixes, unlike ranking and database lookup models. Evaluation scores of Seq2Seq Transformer and pre-trained models (i.e., T5 and BART) indicate that the pre-trained models provide better input representation and perform better. BART + ITC fuses the additional context (top-ranked completions obtained from MPC_{Main}) implicitly with two-step training. However, the results are not promising, indicating that the model's learning is distracted in the two-stage training. Overall, adding explicit context leads to better performance, as shown in BART + MPC_{Main} model.

Eventually, adding context from MPC_{Main} and $\text{MPC}_{\text{Synth}}$ helps the proposed Trie-NLG model perform the best.

The absolute evaluation scores for Unseen AOL data are much higher as compared to Seen AOL data. We observe similar trends for the Bing dataset as well. There could be two possible causes for this: (1) Unseen dataset is ∼11% of the original data, and hence it is much smaller compared to the Seen dataset, and (2) the average prefix lengths for Seen AOL, Unseen AOL, Seen Bing, and Unseen Bing, are 8.1, 20.9, 14.5 and 25.9 respectively. In the Unseen dataset, the lengths of the prefixes are longer compared to Seen, which provides more context and the generative models perform better. Most of the baseline models' performance on the Unseen dataset is very poor, but the proposed Trie-NLG achieves much better performance which shows the promising prospect of our approach. GRM is a ranking model based on clicked queries. As the Unseen dataset does not have click information, GRM models cannot be built.

The MPCMain + MPCSynth model is the best-performing model for the AOL Unseen dataset, and it surpasses the Trie-NLG by a small margin. However, for the Bing Unseen dataset, the proposed model outperformed all the models. There can be multiple possible reasons behind this observation. For example, (1) **Dataset Timeline:** The AOL dataset is from 2006 while Bing data was collected in 2020-21. Pre-trained NLG models (like BART and T5) have been trained with recent corpus whose vocabulary is expected to be better aligned with recent Bing data rather than AOL. Final suggestions from the model for Unseen AOL data are hence governed by only the partially-aligned language model and without any context from the trie. (2) **Query Log Size:** Bing dataset has 20M queries compared to 4M in AOL dataset. This leads to better synthetic suggestions for Bing, in turn leading to better context augmentation for the Bing Trie-NLG model. (3) **Prefix and Session Lengths:** The prefix and session length for Bing (4.434 tokens/prefix and 5.619 queries/session) are longer as compared to AOL (3.061 tokens/prefix and 2.530 queries/session). Longer prefixes and sessions lead to better NLG completions for Bing. Recent search interactions for users do involve longer sessions, and the proposed model is expected to do well in such scenarios.

The overall evaluation results indicate that neither trie nor NLG models are effective individually for such a challenging scenario. The proposed hybrid approach that considers the benefits of both worlds (language semantics from NLG and popularity statistics from trie) through a joint modeling technique is a promising approach and can push the QAC research field forward.

	Prefix Length in [1-5]			Prefix Length in [6-10]			Prefix Length 10+		
Bing Dataset	**ΔMRR**	**ΔBLEU$_{\text{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\text{RR}}$**	**ΔBLEU**	**ΔMRR**	**ΔBLEU$_{\text{RR}}$**	**ΔBLEU**
BART	21.6	-17.6	2.9	38.5	38.4	56.8	49.1	190.0	120.1
BART + MPC$_{\text{Main}}$	52.1	-16.8	10.1	55.2	49.3	72.1	55.6	218.5	145.1
TRIE-NLG	**53.2**	**-15.9**	**11.4**	**56.4**	**50.1**	**73.6**	**60.7**	**221.1**	**149.5**
AOL Dataset	**MRR**	**BLEU$_{\text{RR}}$**	**BLEU**	**MRR**	**BLEU$_{\text{RR}}$**	**BLEU**	**MRR**	**BLEU$_{\text{RR}}$**	**BLEU**
BART	41.3	8.0	35.39	52.2	14.3	53.39	63.2	32.8	75.40
BART + MPC$_{\text{Main}}$	41.5	8.0	35.25	54.1	14.7	54.08	65.1	33.4	76.20
TRIE-NLG	**42.1**	**8.1**	**35.58**	**55.2**	**14.9**	**55.56**	**73.4**	**35.1**	**82.79**

Table 4.6: Performance analysis for short prefixes. For Bing, % improvements over MPC$_{\text{Train}}$+MPC$_{\text{Synth}}$ are reported. For AOL, actual scores are reported.

4.6.2 Performance Analysis for Short Prefixes

Table 4.6 shows the performance of our proposed TRIE-NLG model and two best baselines BART and BART+MPC$_{\text{Main}}$ for different prefix lengths. The evaluation scores are reported for three different buckets based on the character length of the prefix: [1-5], [6-10] and 10+. The evaluation scores in the 10+ bucket are higher as compared to the other two. It indicates that as the prefix length increases, the performance of all the models increases. *It is aligned with the intuition that the model generates more accurate predictions as the prefix becomes longer.* The model performance improves across both datasets for short prefixes as the additional context is added, i.e., BART+MPC$_{\text{Main}}$ performs better as compared to BART. Moreover, when synthetic completions are included further, i.e., TRIE-NLG, it outperforms both the baselines even for very short prefixes. This provides evidence that adding additional trie knowledge does help to increase relevant context for short prefixes. A few of the Δ BLEU$_{\text{RR}}$ scores for Bing are negative for prefix lengths [1-5]. This implies that the performance of the MPC$_{\text{Main}}$+MPC$_{\text{Synth}}$ model is superior to the three models considered, i.e., BART, BART+MPC$_{\text{Main}}$, and TRIE-NLG. Despite this, the lower negative values for TRIE-NLG demonstrate that its performance is better than the other two models. The reason behind this could be attributed to the fact that (1) the MPC$_{\text{Main}}$+MPC$_{\text{Synth}}$ model demonstrates the best performance for the Bing Unseen dataset in terms of unseen prefixes, as discussed in Section 4.6.1, and (2) the Δ BLEU$_{\text{RR}}$ metric takes into account both the BLEU and MRR scores. However, other metrics' results show consistent improvement across all prefix types and both datasets.

4.6.3 Ablation Study

Table 4.7 presents ablation results with different experimental setups. In setups 1 to 3, we have removed session and/or external contexts. The model performs worst

#	Ablation	Bing Dataset			AOL Dataset		
	Criteria	**ΔMRR**	**ΔBLEU$_{\text{RR}}$**	**ΔBLEU**	**MRR**	**BLEU$_{\text{RR}}$**	**BLEU**
1	No (Trie Context + Session)	-29.5	-1.5	43.1	5.7	5.9	30.9
2	No Trie Context	36.7	73.5	91.1	51.9	18.3	61.9
3	No Session	29.3	47.3	73.7	15.4	9.5	40.6
4	TRIE-NLG(1)	44.8	81.3	104.8	32.3	15.1	54.4
5	TRIE-NLG(5)	50.9	82.9	110.0	42.6	17.3	59.6
6	TRIE-NLG(8)	52.9	83.8	111.4	28.5	13.8	51.5
7	TRIE-NLG(3)	**56.8**	**88.3**	**114.5**	**56.5**	**19.3**	**66.6**

Table 4.7: Results of ablation study using different experimental setups. TRIE-NLG(m) means "TRIE-NLG + Up to Top-m Completions". For the Bing dataset, percentage improvements over MPC$_{\text{Train}}$+MPC$_{\text{Synth}}$ baseline are reported. For the AOL dataset, actual evaluation scores are reported.

when both the information are removed (setup 1). Modeling with only session (setup 2) performs better than a model that uses only trie context (setup 3), which shows the importance of the user's previous search query log. However, setups 4 to 7 that use both pieces of information perform even better, indicating the importance of both types of context. Setups 4 to 7 differ from each other in the number of candidate query completions that are used as additional trie context. It is observed that the use of a single top-ranked candidate query results in worse performance, which may be attributed to an inadequate context. Furthermore, incorporating more than three top-ranked queries also results in poor performance. This can be due to two possible factors: (1) the model may become overwhelmed and unable to effectively distinguish relevant information from the trie context in the presence of too many suggestions in the input, or (2) the trie context may become too long, hindering the model's ability to effectively utilize session signals. Overall, TRIE-NLG with top-3 trie candidate completions (i.e., $m = 3$) in the input performs the best. We observe similar trends for both AOL and Bing datasets.

4.6.4 Trie Completion Retention Analysis

Further, we analyze *how many* trie candidates (i.e., completions) are generated as completions by the proposed TRIE-NLG model, and *what position* they appear in. For simplicity, we only consider up to 3 candidate queries and seen test datasets. *In the ideal scenario, the best performing model should retain good candidate queries of MPC$_{Main}$ into the recommended completion list as well as generate new completions.* Table 4.8 shows that ∼19% and ∼43% examples do not retain any completions for Bing and AOL datasets, respectively. At the same time, ∼23% of the Bing examples retain all the input candidate queries. For the AOL dataset, only 3% of examples have all the input candidates; this value is very low because MPC_{Main} is created with only Bing historical search log but used for generating completions for AOL prefixes. So the trie-recommended completions may not be very relevant and hence not considered by Trie-NLG for the AOL dataset.

Tables 4.9 and 4.10 provide the position distribution of each trie candidate in the Trie-NLG output for Bing and AOL datasets respectively. In more than 40% of examples, the top-ranked candidate query from the trie doesn't appear in the final generated output. On the other hand, for 37.6% examples, the top trie candidate is also the top suggestion from Trie-NLG for the Bing dataset. This also indicates that the model does not blindly copy the trie candidates as outputs. Instead, it learns to determine the candidate's goodness or fit for the specific input and performs the generation accordingly. Similar trends have been observed for AOL.

t	0	1	2	3
Bing Seen Test Dataset	19.0%	26.4%	30.5%	23.6%
AOL Seen Test Dataset	43.1%	36.1%	17.2%	3.2%

Table 4.8: Number of examples where t trie suggestions were retained in the Trie-NLG generated completions for *Seen Test Datasets*.

Rank↓/Pos→	1	2	3	4	5	6	7	8	None
1	37.60	10.17	3.92	2.10	1.43	1.15	0.94	0.97	41.69
2	18.50	19.51	7.86	3.81	2.45	1.81	1.53	1.511	41.69
3	9.98	12.50	11.82	7.04	4.49	2.74	1.88	1.67	47.83

Table 4.9: Percentage of times the candidate suggestion from trie was copied to [1-8]th positions ('Pos') as output by Trie-NLG for Bing Seen Test Dataset. 'None' indicates the candidate suggestion was not a part of Trie-NLG output. Results are shown for Seen Test Data when m=3. 'Rank' indicates the rank of candidate suggestion from trie.

4.6.5 Runtime Analysis

Table 4.11 shows inference (generation) times for the three models on an A100 Nvidia GPU. Trie lookups are very cheap compared to BART-based suggestion generation. Hence, our method Trie-NLG has almost similar runtimes compared to a standard BART model.

Rank↓/Pos→	1	2	3	4	5	6	7	8	None
1	25.92	6.81	3.43	2.22	1.57	1.23	1.03	1.12	56.62
2	7.41	5.47	3.26	2.23	1.65	1.29	1.11	1.11	56.62
3	3.97	3.46	2.53	1.97	1.54	1.21	1.04	1.01	83.24

Table 4.10: Percentage of times the candidate suggestion from trie was copied to [1-8]th positions (‘Pos’) as output by Trie-NLG for AOL Seen Test Dataset. ‘None’ indicates the candidate suggestion was not a part of Trie-NLG output. Results are shown for Seen Test Data when m=3. ‘Rank’ indicates the rank of candidate suggestion from trie.

Models	**Bing Test Dataset**			**AOL Test Dataset**		
	Seen	**Unseen**	**Total**	**Seen**	**Unseen**	**Total**
BART	11.25	11.75	11.29	11.69	12.83	11.82
BART+ MPC$_{\text{Main}}$	12.28	12.16	12.27	12.17	13.05	12.27
TRIE-NLG	12.34	12.25	12.34	12.29	13.20	12.40

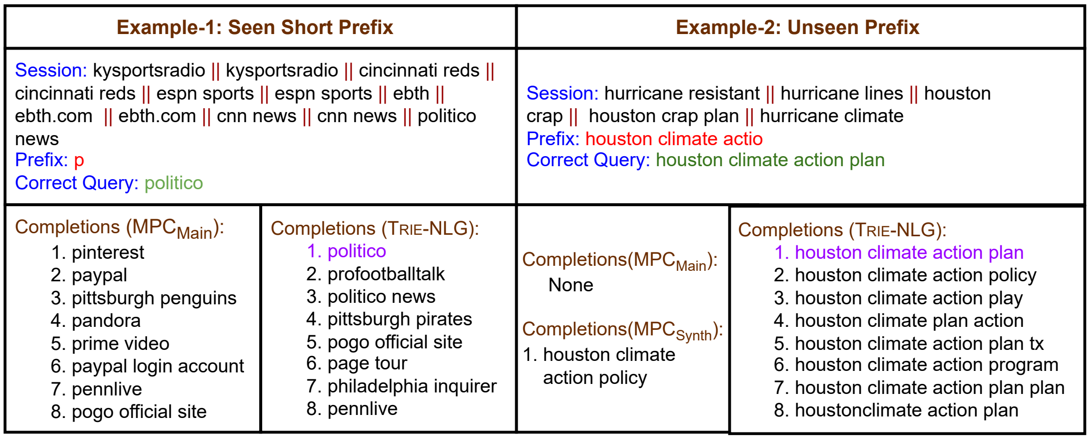
Table 4.11: Runtime of different test dataset splits. Values are in milliseconds(ms)/record for 8 auto-complete generations.

4.6.6 Case Studies

Fig. 4.3 shows two examples of suggestions for a short-seen prefix and an unseen prefix respectively. In the first example, the prefix ‘p’ is very short and MPC$_{\text{Main}}$ is unable to understand the context and recommends more general/popular completions. Whereas the proposed Trie-NLG model learned the personalized context and recommended correct and more relevant completions. The model also considers recommendations from MPC$_{\text{Main}}$ as additional context. For instance, ‘pogo official site’ is present in MPC$_{\text{Main}}$ and recommended by Trie-NLG, although there is no relevant context in the session. In Example-2, there is no query recommendation from MPC$_{\text{Main}}$ for the unseen prefix, but MPC$_{\text{Synth}}$ has one recommendation and that acts as additional context for Trie-NLG. Trie-NLG generates more relevant completions and the top-ranked completion is correct. In summary, we can conclude that the additional trie context is useful for the generative model and helps Trie-NLG to generate more accurate and relevant query completions.

4.7 Conclusion

We proposed Trie-NLG model for personalized QAC. It is based on context augmentation in the NLG model where the additional context is obtained from the main



Example-1: Seen Short Prefix		Example-2: Unseen Prefix	
Session: kysportsradio || kysportsradio || cincinnati reds || cincinnati reds || espn sports || espn sports || ebth || ebth.com || ebth.com || cnn news || cnn news || politico news Prefix: p Correct Query: politico		Session: hurricane resistant || hurricane lines || houston crap || houston crap plan || hurricane climate Prefix: houston climate actio Correct Query: houston climate action plan	

Completions (MPC$_{\text{Main}}$):
1. pinterest
2. paypal
3. pittsburgh penguins
4. pandora
5. prime video
6. paypal login account
7. pennlive
8. pogo official site
	Completions (TRIE-NLG):
	1. politico
	2. profootballtalk
	3. politico news
	4. pittsburgh pirates
	5. pogo official site
	6. page tour
	7. philadelphia inquirer
	8. pennlive
		Completions(MPC$_{\text{Main}}$):
		None
		Completions(MPC$_{\text{Synth}}$):
		1. houston climate action policy
			Completions (TRIE-NLG):
			1. houston climate action plan
			2. houston climate action policy
			3. houston climate action play
			4. houston climate plan action
			5. houston climate action plan tx
			6. houston climate action program
			7. houston climate action plan plan
			8. houstonclimate action plan

Figure 4.3: Sample generations from TRIE-NLG model. Here we consider two examples: a seen short prefix and an unseen prefix.

trie or the synthetic trie. To the best of our knowledge, this is the first study to use the trie context in NLG models for QAC. We primarily focused on solving the problem of short and unseen prefixes. The model was evaluated on a prepared AOL QAC dataset and a real prefix-to-click QAC dataset from Bing. The proposed model outperformed all the baselines while specifically improving the performance for short and unseen prefixes.

4.8 Insights, Limitations and Future Work

Insights: The augmentation of the trie context is performed at several positions, including in the beam search decoding algorithm, in the self-attention of the encoder, in the self-attention of the decoder, as input to the decoder while using teacher-forcing, and many more. However, we observed that the model performs best when the trie context is augmented in the input, as in the TRIE-NLG model.

Limitations: The proposed TRIE-NLG model operates in a two-step process, comprising the extraction of auto-completions from a trie and augmentation in the NLG model. As a consequence of this approach, the model exhibits slightly higher latency compared to the standard NLG model due to trie lookup. However, the trie lookup time is notably low. It is crucial to highlight that the proposed model is built upon a pre-trained NLG model (BART), which renders it susceptible to displaying unexpected outcomes inherited from the pre-training phase [GGS+20]. Such outcomes may include toxicity, bias, hallucination, misinformation, and other similar issues.

Future Work: Not all session queries are relevant to the current user prefix. Irrelevant session queries lead to noisy training data. In the future, we plan to explore modeling techniques that can select and encode only the relevant queries as personalized contexts for TRIE-NLG. Additionally, we will explore *on-the-fly* models instead of two-step as in TRIE-NLG. Exploring more novel RAG [LPP+20] modeling, which can provide better context/completions instead of trie completions to boost the performance, will be an interesting extension of the work. Finally, we will explore a transfer learning approach to extend this to a multilingual QAC system.

Chapter 5

Mitigating Catastrophic Forgetting to Enable Zero-Shot Cross-Lingual Generation

5.1 Introduction

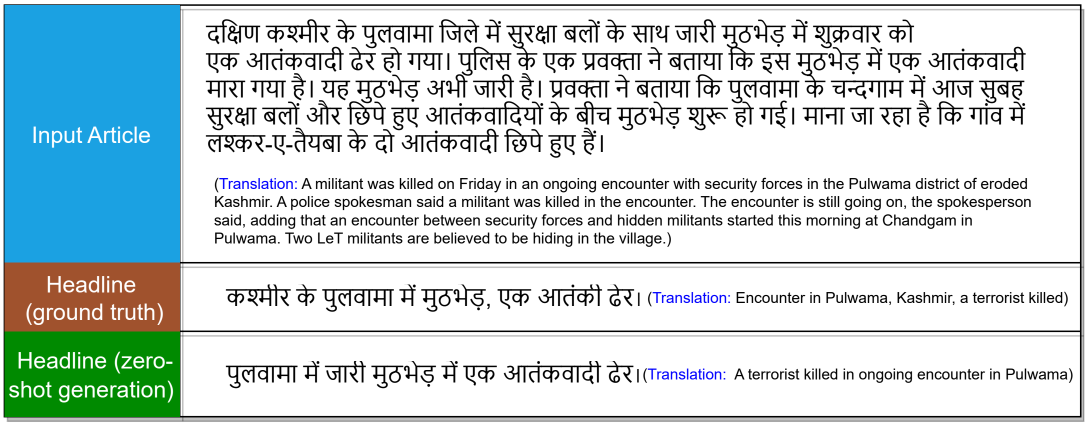
In the first part of the thesis, our focus was on advancing generative NLP by generating diverse text (Chapter 3) and mitigating issues related to limited context (Chapter 4). We made efforts to enhance NLG modeling for two specific applications (distractor generation and PQAC), employing traditional LSTM-based and modern large language model (LLM)-based approaches, respectively. In the second part of the thesis (i.e., the next three chapters), we delve into novel modeling frameworks to extend these technologies to *limited data* scenarios, a common occurrence in low-resource languages (LRLs). Here, we focus on NLG modeling in zero-shot and few-shot settings, which enable scalability. The next three chapters can be read independently. We continue to explore LLM-oriented modeling.

The deep learning-based modeling for NLP heavily relies on a large amount of labeled training data. Such labeled data is publicly available for high-resource languages (HRLs), i.e., English. However, challenges arise when modeling with limited labeled data, which is often the case for LRLs like Hindi, Japanese, and others. The scarcity of labeled data for LRLs is more pronounced for NLG tasks, as task-specific data availability for LRLs is more rare. Manually annotating large task-specific datasets is a time-consuming, expensive, and uninteresting process, which hampers model development and product deployment for LRLs. One promising direction is cross-lingual modeling [HRS+20, CKG+20, LLG+20a], which involves training a model on a large high-resource task-specific language (generally English) and zero-shot or few-shot inference in LRLs. This approach leverages *supervision transfer* from HRLs to LRLs and enables technology for unseen LRLs (zero-shot) or LRLs with limited training examples (few-shot). While cross-lingual modeling has gained attention for natural language understanding (NLU) tasks, the field of cross-lingual generation remains relatively under-explored due to several additional challenges.

To understand these challenges, let's consider a concrete cross-lingual abstractive text summarization (ATS) task. *It is a task of generating a grammatically coherent, semantically correct, and abstractive summary given an input article.* A typical zero-shot cross-lingual generation model involves two main steps: *(1) Training with HRLs:* Train (fine-tune) a model (LLM) using a large labeled dataset from HRLs, typically English. For instance, training with an English ATS dataset and *(2) Zero-shot generation in LRLs:* Utilize the trained model for zero-shot generation in target LRLs. For instance, when given input in an LRL (e.g., a Hindi article), the model generates a summary in the same LRL (Hindi summary). Unlike NLU tasks, in the zero-shot cross-lingual generation, the text needs to be generated in the target LRL, which generally suffers from Catastrophic Forgetting[1] (CF; [VdVT19]) problem. Due to CF, the model generates text in fine-tuned HRL (e.g., English) or produces code-mixed output with both fine-tuned HRL and target LRL. *Towards mitigating the catastrophic forgetting problem and improving cross-lingual supervision transfer, we propose a novel unsupervised modeling framework called* **ZmBART** [MDKD21b]. This framework is designed to facilitate well-formed zero-shot generation for LRLs.

We carefully selected four challenging NLG tasks, i.e., news headline generation (NHG), question generation (QG), abstractive text summarization (ATS), and distractor generation (DG) to evaluate the proposed ZmBART model performance. NHG and ATS require understanding input passage to generate meaningful headlines and summaries, respectively. QG task should contextualize information from a passage and answer to generate high-quality questions. Distractor generation is the task of generating incorrect options from reading comprehension MCQ. It is challenging because generated distractors should be in the question's context but not semantically equivalent to the answer. Further, we consider two LRLs, i.e., Hindi and Japanese, from two different language families. English is selected as the HRL from which the learned supervision would be transferred to the LRLs. All three selected languages are different in their syntactic structures and typologically diverse. This will test

[1] also known as Accidental Translation [XCR+21] or off-target problem



Input Article	दक्षिण कश्मीर के पुलवामा जिले में सुरक्षा बलों के साथ जारी मुठभेड़ में शुक्रवार को एक आतंकवादी ढेर हो गया। पुलिस के एक प्रवक्ता ने बताया कि इस मुठभेड़ में एक आतंकवादी मारा गया है। यह मुठभेड़ अभी जारी है। प्रवक्ता ने बताया कि पुलवामा के चन्दगाम में आज सुबह सुरक्षा बलों और छिपे हुए आतंकवादियों के बीच मुठभेड़ शुरू हो गई। माना जा रहा है कि गांव में लश्कर-ए-तैयबा के दो आतंकवादी छिपे हुए हैं। (Translation: A militant was killed on Friday in an ongoing encounter with security forces in the Pulwama district of eroded Kashmir. A police spokesman said a militant was killed in the encounter. The encounter is still going on, the spokesperson said, adding that an encounter between security forces and hidden militants started this morning at Chandgam in Pulwama. Two LeT militants are believed to be hiding in the village.)
Headline (ground truth)	कश्मीर के पुलवामा में मुठभेड़, एक आतंकी ढेर। (Translation: Encounter in Pulwama, Kashmir, a terrorist killed)
Headline (zero-shot generation)	पुलवामा में जारी मुठभेड़ में एक आतंकवादी ढेर।(Translation: A terrorist killed in ongoing encounter in Pulwama)

Figure 5.1: Sample zero-shot news headline generation with ZmBART in the Hindi language

the effectiveness of the proposed model. As there is no established publicly available dataset for DG in Hindi, we also create a new high-quality DG dataset for Hindi called **HiDG**[2].

The ZmBART is developed on top of mBART [LGG+20c], a multilingual pre-trained language model trained with 25 languages with denoising objectives (masking and sentence permutation). We perform adaptive unsupervised training by *further* pre-training mBART with a novel *auxiliary task*. Then, this trained model is fine-tuned on large task-specific supervised data in English and evaluated directly with Hindi and Japanese languages in zero and few-shot settings. The auxiliary task is critical to mitigate CF and improve the cross-lingual supervision transfer. This framework can be directly applied to multiple cross-lingual generation tasks without modifying any hyper-parameters values. Fig. 5.1 shows a sample zero-shot output generation for the NHG task with the ZmBART.

Our main contributions through this work can be summarized as follows:

1. We propose a novel zero-shot cross-lingual transfer and generation framework called ZmBART, which does not require parallel data/pseudo-parallel and without back-translated data. It is scalable to multiple NLG tasks without even modifications in hyper-parameter values.

2. The ZmBART is powered by adaptive pre-training with a navel auxiliary task as a learning objective. This helps to mitigate catastrophic forgetting problems and generates well-formed zero-shot text in target LRLs.

[2] HiDG dataset download link: https://github.com/kaushal0494/ZmBART

3. We demonstrate the effectiveness of ZmBART on four cross-lingual generation tasks across three typologically diverse languages.

4. We have created HiDG, a high-quality distractor generation dataset for the Hindi language.

5.2 Related Work

Early works on cross-lingual generation rely on machine translation (MT). Wan et al. [WLX10a] leveraged the MT pipeline for cross-language ATS. They first translate the non-English test instances into English. This translated text is fed through the trained, supervised model (with document ATS data in English) to generate English summaries. Finally, these summaries are translated back to the target language. Shen et al. [SCY+18] and Duan et al. [DYZ+19] used MT systems to generate pseudo-training data for cross-lingual ATS and NHG, respectively. However, these MT-based models are not suitable for LRLs as they do not share parameters across languages, making them not scalable. Furthermore, translations are not perfect, leading to the propagation of translation errors.

The ZmBART work was carried out in the time span when PLMs were emerging; there have been limited efforts in the direction of supervision transfer from HRL(s) to LRL(s) for language generation tasks. Kumar et al. [KJM+19a] used back-translation (needs MT system) and annotated supervised data for cross-lingual question generation. Chi et al. [CDW+20a] used parallel data to train a sequence-to-sequence model for zero-shot cross-lingual abstractive text summarization and question generation. Lewis et al. [LGG+20b] consider the task of ATS where small annotated data is available in multiple languages. The model is first pre-trained with mono-lingual paragraphs. Then, this model is fine-tuned with the small ATS dataset of all the languages except the test language. The final model is used for zero-shot ATS in test languages. These fine-tuning and testing steps are repeated for different languages. This is similar to the k-fold cross-validation setup. This approach needs annotated data in multiple languages and other existing supervision transfer methods require parallel data for cross-lingual tasks. Either they use available parallel corpora directly or translate/ back-translate data as pseudo-parallel data. Both these approaches pose significant challenges, as parallel data for multiple languages is difficult to obtain. MT systems (to obtain pseudo-parallel data) are far from perfect or unavailable for many LRLs.

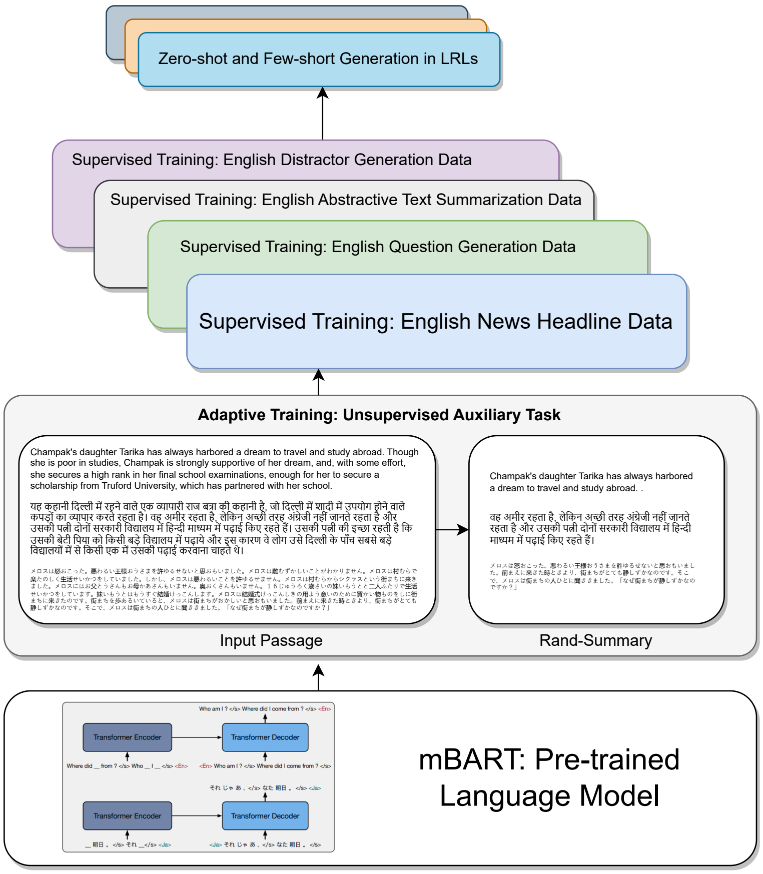
Unlike the previous approaches, we did not use any parallel data or back-translation in our proposed framework. We did not pre-train any model from scratch. Instead, we leveraged the existing pre-trained multilingual language model, i.e., mBART. We included four challenging generation tasks across three typologically diverse languages. We did not modify any hyper-parameters across the tasks and languages. All these considerations make the framework simple and easy to use. Further, it enables adding different languages and NLG tasks in the proposed framework as a simple extension exercise.

5.3 Methodology

The proposed ZmBART model has two-fold objectives: (i) Mitigating the effect of catastrophic forging and well-formed zero-shot generation in LRLs and (ii) Improving the cross-lingual transfer singles from HRLs to LRLs for model performance boost. To achieve these objectives, three novel modeling aspects are adapted in ZmBART. (1) We perform an adaptive *further pre-training* of mBART with a novel auxiliary task. The auxiliary task is designed in such a way that the objective function of the auxiliary task is close to fine-tuning tasks and only utilizes the mono-lingual data from the considered languages. (2) we freeze the model components while fine-tuning with task-specific HRLs which contextualize previous learning and help to mitigate the CF issue. (3) We modified the mBART language identifier tag as `<fxx><2xx>` in the input data instance where `<xx>` indicates the ISO-2 language code. Given an input sentence and the language tag, the model encodes the sentence in multi-lingual space. By conditioning on the encoded representation and language tag the decoder generates output text in the target language. The proposed model is developed on top of the base pre-trained mBART [LGG$^+$20c] model and does inference in zero-shot and few-shot settings. Figure 5.2 shows an overview of the proposed ZmBART framework. Next, we will begin by providing a brief overview of the mBART model. Following that, we will delve into the three modeling components, and finally, we will cover the training and generation details.

5.3.1 Background: Multilingual BART (mBART)

Multilingual BART (mBART) [LGG$^+$20c] is an extension of the BART model [LLG$^+$20b] designed to work with multiple languages. It is a transformer-based sequence-to-sequence (aka. encoder-decoder) pre-trained model. The model is trained on mono-


Zero-shot and Few-short Generation in LRLs
Supervised Training: English Distractor Generation Data
Supervised Training: English Abstractive Text Summarization Data
Supervised Training: English Question Generation Data
Supervised Training: English News Headline Data
Adaptive Training: Unsupervised Auxiliary Task
Input Passage
Rand-Summary
Transformer Encoder
Transformer Decoder
Who am I ? </s> Where did I come from ? </s> <En>
Where did __ from ? </s> Who __ I __ </s> <En>
<En> Who am I ? </s> Where did I come from ? </s>
それ じゃ あ 、</s> なた 明日 。 </s> <Ja>
__ 明日 。 </s> それ __</s> <Ja>
<Ja> それ じゃ あ 、</s> なた 明日 。 </s>
mBART: Pre-trained Language Model

Champak's daughter Tarika has always harbored a dream to travel and study abroad. Though she is poor in studies, Champak is strongly supportive of her dream, and, with some effort, she secures a high rank in her final school examinations, enough for her to secure a scholarship from Truford University, which has partnered with her school.

यह कहानी दिल्ली में रहने वाले एक व्यापारी राज बत्रा की कहानी है, जो दिल्ली में शादी में उपयोग होने वाले कपड़ों का व्यापार करते रहता है। वह अमीर रहता है, लेकिन अच्छी तरह अंग्रेजी नहीं जानते रहता है और उसकी पत्नी दोनों सरकारी विद्यालय में हिन्दी माध्यम में पढ़ाई किए रहते हैं। उसकी पत्नी की इच्छा रहती है कि उसकी बेटी पिया को किसी बड़े विद्यालय में पढ़ाये और इस कारण वे लोग उसे दिल्ली के पाँच सबसे बड़े विद्यालयों में से किसी एक में उसकी पढ़ाई करवाना चाहते थे।

メロスは怒おこった。悪わるい王様おうさまを許ゆるせないと思おもいました。メロスは難むずかしいことがわかりません。メロスは村むらで楽たのしく生活せいかつをしていました。しかし、メロスは悪わるいことを許ゆるせません。メロスは村むらからシクラスという街まちに来きました。メロスにはお父とうさんもお母かあさんもいません。奥おくさんもいません。１６じゅうろく歳さいの妹いもうとと二人ふたりで生活せいかつをしています。妹いもうとはもうすぐ結婚けっこんします。メロスは結婚式けっこんしきの用よう意いのために買かい物ものをしに街まちに来きたのです。街まちを歩あるいていると、メロスは街まちがおかしいと思おもいました。前まえに来きた時ときより、街まちがとても静しずかなのです。そこで、メロスは街まちの人ひとに聞ききました。「なぜ街まちが静しずかなのですか？」

Champak's daughter Tarika has always harbored a dream to travel and study abroad. .

वह अमीर रहता है, लेकिन अच्छी तरह अंग्रेजी नहीं जानते रहता है और उसकी पत्नी दोनों सरकारी विद्यालय में हिन्दी माध्यम में पढ़ाई किए रहते हैं।

メロスは怒おこった。悪わるい王様おうさまを許ゆるせないと思おもいました。前まえに来きた時ときより、街まちがとても静しずかなのです。そこで、メロスは街まちの人ひとに聞ききました。「なぜ街まちが静しずかなのですか？」

Figure 5.2: Overview of ZmBART developed on top of mBART [LGG+20c] model

lingual data from 25 languages obtained from the Wikipedia Common Crawl corpus using the BART language model objective. Specifically, the training data is a concatenation of data from K languages, denoted as $\mathcal{D} = \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$, where $\mathcal{D}_i$ represents a collection of monolingual documents in language i. They introduced two types of noise to corrupt the text: (1) random token span masking and (2) sentence order permutation. mBART is trained as a denoising autoencoder. During training, the task is to predict text X from its corrupted version $g(X)$, where g represents the noise function. The objective is to maximize the following function:

$$\mathcal{L}_\theta = \sum_{\mathcal{D}_i \in \mathcal{D}} \sum_{x \in \mathcal{D}_i} log P(x|g(x);\theta), \tag{5.1}$$

Here, x is a data instance in language i, θ is model parameters, and the probability distribution P is defined by the sequence-to-sequence model. mBART has achieved state-of-the-art results in a sentence and document-level machine translation tasks. Further details about the mBART model can be found in [LGG+20c].

5.3.2 Unsupervised Auxiliary Task

Although the mBART pre-trained model encodes a common multilingual latent representation space, it can not be used directly for cross-lingual generation. Because the model is jointly trained on denoising objectives that do not directly follow auto-regressive decoding, thereby causing a mismatch between pre-training and fine-tuning objectives [CDW+20a, DCLT19]. To overcome this problem, an unsupervised auxiliary task is introduced. We design the auxiliary task with the following desiderata in mind: (1) It should only utilize monolingual data from considered languages, (2) aid in mitigating CF/AT issues, (3) It should enhance the latent representation space for considered languages, and (4) maintain close proximity between the auxiliary task objective and downstream NLG tasks objective.

The downstream NLG tasks considered in this work are expected to retain/-generate words from the input. Motivated by this property, we propose a novel auxiliary task that encodes the input and utilizes this encoded representation to generate partial input as output in an auto-regressive manner. Specifically, the auxiliary task is designed as: *Given an input Passage, generate a few random sentences (called Rand-Summary) from the passage.* It is language-independent and scalable. Through empirical analyses, we found that randomly selecting 20% of the sentences from the input passage as a target gives the best results. Furthermore, we constrained the input length to be between 5 and 25 sentences and the output comprising 1 to 5 random sentences from the passage. We collected an equal proportion of small monolingual datasets (∼11K examples from each language) from all languages considered. Table 5.1 outlines the data preparation steps for the auxiliary task.

1. Generate a random number $k \in \{5, 25\}$. k denotes the size of the input passage.

2. Passage: Append k continuous sentences, starting from a random index of monolingual corpus D_i of the i^{th} language.

3. Rand-Summary: Randomly select 20% of the sentences from the passage.

4. Repeat steps 1 to 3 for all p languages.

5. Repeat steps 1 to 4 for N times to collect $Np <$ passage, rand-summary $>$ pairs.

Table 5.1: Data preparation steps for the Auxiliary Task

The auxiliary task is an additional pre-training step (aka. adaptive training) for *better warm-start* to downstream auto-regressive NLG tasks - although the downstream tasks (Distractor/Question/Summary/Headline generation) can be different from the auxiliary task. Additionally, this step allows the model to have a closer look at the languages under consideration and enrich/adjust the representations and parameters accordingly.

5.3.3 Freezing Model Components

During supervised training, while fine-tuning ZmBART with task-specific HRL data, we freeze all word embeddings and the parameters of the decoder layers[3]. This approach is adapted to ensure that ZmBART's previous learning (multilingual petraining and adaptive training) and latent space are not overwritten during supervised training.

5.3.4 Adding Language Tag

We have modified the mBART model's language tag for the cross-lingual generation framework. We concatenate `<fxx><2xx>` tag in the source side of the training data, where `<xx>` is the ISO-2 language code. The language tag acts as a flag to trigger the zero-shot generation in target `<xx>` languages.

The ablation study (Section 5.6.3) provides evidence that all three components are necessary to mitigate CF/AT problems effectively and enable well-formed zero-shot text generation in LRLs.

[3] We have experimented with freezing different model components - proposed setup works best.

5.3.5 Model Training and Generation

We consider four tasks: Question Generation (QG), News Headline Generation (NHG), Abstractive Text Summarization (ATS), and Distractor Generation (DG), in three typologically diverse languages. The HRL is English (en), and the LRLs are Hindi (hi) and Japanese (ja). First, the mBART model undergoes adaptive pre-training with the auxiliary task to obtain the ZmBART model. Then, for each NLG task, the ZmBART model is fine-tuned using task-specific HRL data (English). During task-specific fine-tuning, the word embeddings and all the decoder parameters are frozen. This allows the model to enable cross-lingual transfer (i.e., supervision transfer from HRL) and contextualize previous pre-training and auxiliary fine-tuning. The final model is used for zero-shot and few-shot generation (with 1000 examples) in LRLs. The language tag is added to the source input during all fine-tuning and generation steps. The mBART model, fine-tuned with the proposed auxiliary task and HRL of downstream NLG task, is referred to as the ZmBART model.

5.4 Experimental Setup

With the ZmBART, we aim to address the following research questions: (1) Does the ZmBART successfully mitigate the CF/AT problem? (2) How does the ZmBART perform compared to existing literature baselines? (3) Does the model's performance persist across different tasks and LRLs? and (4) Does the model's performance improve with few-shot training? Considering these questions, we have designed the following experimental setup:

5.4.1 Baselines

To compare the ZmBART model performance, we have developed three strong baselines. Details of these baselines are mentioned below:

- MTPipeline: We fine-tune mBART on task-specific English data. Then, the input of non-English test data is first translated into English and passed to the fine-tuned model to generate the output. Finally, the output is translated back to the non-English language. Google Translator is used for translations.

- MonoMask: This is similar to the ZmBART model. Here, we use *word masking* objective instead of auxiliary task objective. This is inspired by the success of the BERT model [DCLT18]. Following BERT, we randomly mask 15% words from each input sentence. Given marked input passage model has to generate the correct passage. The aim is to test the effectiveness of the proposed auxiliary task.

- ParaMask: Drawing inspiration from the [CDW+20a], we have considered Hindi-English and Japanese-English parallel data in this baseline. For each data, we concatenated parallel instances together and treated them as monolingual data. This was applied to both data, and the resulting two monolingual data (corresponding to two parallel data) were merged into a single final data. Finally, we perform a word masking objective similar to MonoMask baseline model with final data. Including parallel data provides explicit cross-lingual supervision transfer and is expected to boost the model performance.

5.4.2 Evaluation Metrics

We employ both automated and human evaluation metrics for performance comparison. Multiple metrics are used in the literature for NLG tasks. Here, we consider commonly used metrics by the research community. For automatic evaluation, we used both lexical match metrics (**BLEU**[4] [PRWZ02b] and **ROUGE**[5] [Lin04a]) as well as embedding-based (semantic-based) evaluation metric (**BERTScore**[6] [ZKW+20]). Specifically, for QG and DG tasks, we employ the BLEU-4 (BL), ROUGE-L (R-L), and BERTScore (BS) metrics, and for ATS and NHG tasks, we rely on ROUGE-1, ROUGE-2, and ROUGE-L metrics.

We follow a similar approach for human evaluation as Chi et al. [CDW+20a]. We sampled 50 generated data points each from QG, ATS and NHG tasks in both Hindi and Japanese languages. We use three human evaluation metrics: *Fluency* (Flu), *Relatedness* (Rel) and *Correctness* (Corr). **Fluency** measures *how fluent the generated text is*. **Relatedness** indicates *how much the generated outputs are in the context with input(s)*, **Correctness** measures *semantics and meaningfulness*. For DG, we use an additional metric called **Distractibility** that measures *the degree of confusion for generated incorrect options*. For the DG task, there can be a large number of good distractors for a given input; in such a situation, the manual evaluation is more reliable. We sample large generated outputs (100 generations) for the DG task. We employed a large pool of evaluators from native Hindi and Japanese speakers to

[4] https://github.com/mjpost/sacrebleu
[5] https://github.com/pltrdy/files2rouge
[6] https://github.com/Tiiiger/bert_score

evaluate Hindi and Japanese output texts, respectively. We asked each annotator to rate the generated texts on a scale of 1-5 (1 is very bad and 5 is very good) for all the metrics. We intentionally selected three models,i.e., outputs from ZmBART and the two best baselines, to reduce the evaluator's workload.

5.4.3 Implementation Details

We use mBART as a `base` multilingual pre-trained model, which is a standard sequence-to-sequence Transformer architecture with 12 layers (each 16 heads). The model has a dimension of 1024 (approx 680M parameters). Additional layer-normalization was used with both the encoder and decoder. We found that FP16 precision stabilized the training. We trained all the models on 4 Nvidia V100 GPUs (32GB). We use the Adam optimizer ($\epsilon = 1e^{-6}$, $\beta_2 = 0.98$) and linear learning rate decay scheduling. The training started with a dropout value of 0.3 and was later reduced to 0.2 after 20k steps and 0 after 40k steps. The loss function was cross-entropy label smoothing loss. 2500 warm-up steps and $3e^{-5}$ learning rate were used. The model selection was done based on validation data likelihood. We use beam-search with beam size 5 in the decoding for all the tasks. We use `mBARTCC25` as a base pre-trained checkpoint.

The above set of hyper-parameters is used for all the downstream NLG tasks as well as the auxiliary task. We process different batch sizes of input for different tasks. We use 2048, 3000, 4096, 2048, and 5000 tokens per GPU for ATS, DG, QG, auxiliary, and NHG tasks. We use shared Byte Pair Encoding (BPE) vocabulary from a sentence-piece tokenizer of size 250k. We use 34k/1k/1k (train/validation/test) data points for auxiliary language (approx 11333 from each language). We train the mBART model with the auxiliary task around 10k steps. Training time for the auxiliary task is around 2-3 hours. The fine-tuning times for TS, QG, NHG, and DG were around 4-5, 1-2, 1-2, and 2-3 hours. We observe a longer fine-tuning time for ATS because of long passages. We selected the best model based on loss and perplexity on the validation datasets. We checked with early-stopping and other checkpoints, which resulted in poor performance. For English, Hindi and Japanese, we sacreBLEU[7], Indic-NLP[8] and Kytea[9] tokenizers, respectively.

[7] https://github.com/mjpost/sacrebleu

[8] https://anoopkunchukuttan.github.io/indic_nlp_library/

[9] http://www.phontron.com/kytea/

5.5 Downstream NLG Tasks and Results

In this section, we present details of downstream NLG tasks, share dataset details, and highlight the major results and findings. We consider four tasks: Question Generation (QG), News Headline Generation (NHG), Abstractive Text Summarization (ATS), and Distractor Generation (DG), in three typologically diverse languages. The HRL is English (en), and the LRLs are Hindi (hi) and Japanese (ja). Automated evaluation results are presented in Tables 5.2 and 5.3 for Hindi and Japanese, respectively. Human evaluation results for Hindi and Japanese are presented in Tables 5.4 and 5.5, respectively.

5.5.1 News Headline Generation (NHG)

It is a task of *generating grammatically coherent, semantically correct, and abstractive headline, given a news article.* We use 500k/30k/30k (train/validation/test) English NHG data splits from *Gigaword* headline generation corpus[10]. For Hindi and Japanese, we use 1k/1k/5k splits from Kaggle[11] and Iwama et al. [IK19], respectively. Further, we did manual verification to ensure the quality.

The MonoMask baseline is the best among the others, which shows that masking and denoising with monolingual data indeed enrich the multilingual latent space and lead to improved performance. However, ZmBART outperforms the MonoMask model with an absolute difference of 5.22 in the ROUGE-L score, showing the impressive performance of the ZmBART model. Moreover, MonoMask generates code-mixed (Hindi-English or Hindi-Japanese) output in the zero-shot setting. Few-shot training corrects the mistakes of zero-shot models and generates higher-quality output. Despite having explicit cross-lingual information in ParaMask through parallel data, the model performs poorly. One possible reason could be the misalignment of sentences as they are concatenated in a sequential manner. All the baselines and the ZmBART model outperform the MTPipeline model, which indicates the importance of auxiliary tasks. These scores correlate with automated scores, validating ZmBART's genuine performance gain for the NHG task.

[10] https://github.com/harvardnlp/sent-summary

[11] https://www.kaggle.com/disisbig/hindi-text-short-summarization-corpus

Model	News Headline Generation			Question Generation			Abstractive TS			Distractor Generation		
	R-1	**R-2**	**R-L**	**BL**	**R-L**	**BS**	**R-1**	**R-2**	**R-L**	**BL**	**R-L**	**BS**
Zero-shot results												
MTPipeline	16.61	4.91	15.83	2.6	21.31	71.53	11.15	3.11	10.93	1.6	9.66	67.35
MonoMask	29.32	16.36	27.52	3.9	23.70	73.76	18.25	4.92	16.10	2.8	15.86	72.26
ParaMask	24.02	13.41	23.29	4.3	25.29	73.74	10.47	2.55	12.30	2.9	15.43	72.89
ZmBART	**34.94**	**19.38**	**32.74**	**4.4**	**26.51**	**74.19**	**21.27**	**5.30**	**17.64**	**4.1**	**21.05**	**73.39**
Few-shot results (with 1000 supervised data points)												
ZmBART	52.37	35.52	50.50	7.6	34.11	78.29	36.29	14.21	27.22	6.5	26.58	78.27

Table 5.2: Zero-shot and few-shot results for Hindi language

Model	News Headline Generation			Question Generation			Abstractive TS		
	R-1	**R-2**	**R-L**	**BL**	**R-L**	**BS**	**R-1**	**R-2**	**R-L**
Zero-shot results									
MTPipeline	13.82	0.38	7.92	8.9	26.92	71.93	17.90	3.98	18.46
MonoMask	33.75	8.12	17.78	16.6	34.80	74.01	28.74	9.01	23.63
ParaMask	31.58	6.98	18.95	18.2	36.22	74.99	19.17	4.89	18.22
ZmBART	**35.25**	**9.24**	**19.92**	**18.8**	**38.74**	**75.91**	**36.60**	**15.26**	**29.85**
Few-shot results (with 1000 supervised data points)									
ZmBART	47.06	22.36	31.55	30.4	53.98	82.66	41.65	20.33	33.49

Table 5.3: Zero-shot and few-shot results for Japanese language

Model	News Headline Generation			Question Generation			Abstractive TS			Distractor Generation		
	Flu	**Rel**	**Corr**	**Flu**	**Rel**	**Corr**	**Flu**	**Rel**	**Corr**	**Flu**	**Rel**	**Dist**
Annotator set-01												
MonoMask	3.86	4.34	3.94	2.66	3.38	3.52	3.56	3.58	3.22	3.61	4.08	2.89
ParaMask	2.54	2.96	2.28	3.1	3.4	3.78	2.26	2.62	1.92	2.42	3.72	3.08
ZmBART	**4.14**	**4.22**	**4.04**	**3.24**	**3.44**	**3.9**	**4.02**	**4.12**	**3.54**	**4.12**	**4.19**	**3.83**
Annotator set-02												
MonoMask	3.84	4.18	3.8	3.83	4.63	3.96	3.38	3.96	3.4	3.38	3.00	2.24
ParaMask	2.96	3.02	2.7	**3.98**	4.70	3.98	2.96	3.16	2.84	2.97	3.11	**2.46**
ZmBART	**4.12**	**4.38**	**4.16**	3.95	**4.80**	**4.27**	**4.24**	**4.52**	**4.38**	**3.56**	**3.18**	2.36
Annotator set-03												
MonoMask	3.56	3.74	**3.78**	2.68	3.76	3.32	2.9	3.34	2.9	3.96	3.74	3.12
ParaMask	3.1	3.42	2.91	2.80	3.88	3.56	2.64	2.34	2.46	4.13	3.74	2.94
ZmBART	**3.70**	**3.84**	3.76	**2.86**	**4.04**	**3.76**	**4.06**	**3.56**	**3.56**	**4.44**	**4.12**	**3.12**

Table 5.4: Human evaluation results for zero-shot generated outputs in the Hindi language

Model	News Headline Generation			Question Generation			Abstractive TS		
	Flu	**Rel**	**Corr**	**Flu**	**Rel**	**Corr**	**Flu**	**Rel**	**Corr**
Annotator set-01									
MonoMask	2.66	2.98	2.50	1.98	**3.70**	**3.18**	3.04	3.55	3.44
ParaMask	2.26	2.70	2.04	2.00	3.38	2.82	1.44	2.22	2.20
ZmBART	**3.60**	**4.02**	**3.50**	**2.12**	3.30	2.94	**4.24**	**3.90**	**3.90**
Annotator set-02									
MonoMask	2.1	2.58	1.98	1.24	1.70	1.33	2.56	3.40	2.62
ParaMask	1.58	1.78	1.46	**1.46**	1.72	1.78	1.00	1.00	1.00
ZmBART	**3.78**	**4.16**	**3.86**	1.26	**1.76**	**1.88**	**4.04**	**4.26**	**3.84**
Annotator set-03									
MonoMask	2.24	2.72	2.24	**2.34**	2.46	2.39	2.82	3.18	3.52
ParaMask	1.9	2.14	1.82	2.10	2.66	2.28	1.16	1.84	1.44
ZmBART	**2.88**	**3.22**	**2.92**	2.10	**2.70**	**2.46**	**3.32**	**3.52**	**3.04**

Table 5.5: Human evaluation results for zero-shot generated outputs in the Japanese language

5.5.2 Question Generation (QG)

In the Question Generation task, *given an input passage and an answer, the aim is to generate semantically and syntactically correct questions that can produce the answer.* We use SQuAD 1.1 [RZLL16b] English data for supervised training. SQuAD is a popular question and answering (Q&A) dataset consisting of 100k+ <passage, question, answer> tuples. Following [ZNDK18], we split it as 80k/8k/10k training/-validation/test sets. For Hindi we use 1k/5.5k (train/test) combined from MLQA [LOR+20a] and TyDiQA-GoldP [CCC+20a] datasets. We use 1k/1k/5k for Japanese data from Takahashi et al. [TSKK19a]. Hindi and Japanese data are available in SQuAD data format which maintains consistency in terms of passage size, question, and different number of answers. For a given passage and question we randomly sample one answer from the corresponding answer set. We combine the answer and passage as a single input sequence separated by a special token <s>.

Even without any parallel data, ZmBART consistently outperformed all the baselines across all automated evaluation metrics in the zero-shot setting. Regarding manual evaluations, we observed that zero-shot Hindi question generations received high scores from the annotators, while the questions generated for the Japanese language were considered of lower quality. Upon closer inspection of the generated text, we noticed that several zero-shot generated questions in both Hindi and Japanese languages began with English *wh-words*. This mixing of English *code* is possible because these languages are typologically different compared to English. Moreover, the practice of code-mixing between Hindi and English is becoming increasingly common, and annotators generally accept code-mixed Hindi texts. However, such mixing is less common in Japanese text. Consequently, the annotators assigned lower scores to the Japanese-generated texts.

We then attempted to understand the reason for the occurrence of *wh-words* at the beginning of the Hindi and Japanese generations. In English, interrogative sentences often begin with *wh-words*, and the model exposed to these specific characteristics of English interrogative sentences during the English fine-tuning. In zero-shot settings, this exposure impacts the output in other languages, resulting in code-mixed sentences that start with *wh-words*. However, the model effectively captures the semantics of the text, as evidenced by the high BERTScore, indicating a strong cross-lingual transfer of semantic knowledge.

5.5.3 Abstractive Text Summarization (ATS)

In Abstractive Text Summarization, we aim to *generate a grammatically coherent, semantically correct, and abstractive summary given an input document*. We use the WikiLingua [LDCM20b] cross-lingual abstractive summarization dataset containing data available in 18 languages. Prior splits are not available for this dataset, so we created 131k/5k/5k (train/validation/test) splits for English. For Hindi and Japanese, 1k/1k/5k splits were used.

By skimming through data in Hindi, we observe that many input documents consist of technical instructions on the usage of software/tools. Summarizing these instructions is challenging. Zero-shot ZmBART performed better as compared to baselines as shown in human evaluation (Tables 5.4 and 5.5 for Hindi and Japanese, respectively). The human evaluation results correlate with automated evaluation as shown in Tables 5.2 and 5.3. Takahashi et al. [LDCM20b] reported cross-lingual ATS scores with the same data for four different languages. They used the supervised training setup. The R-L scores for the four languages are 34.06, 37.09, 31.67, and 32.33. We obtained few-shot R-L scores of 27.22 and 33.49 for Hindi and Japanese, respectively. While these scores are not directly comparable, they provide a rough estimate of the few-shot performance with the supervised model, which is considered acceptable.

5.5.4 Distractor Generation (DG)

The final task to evaluate the ZmBART's performance is the Distractor Generation. It is the task of generating incorrect options (also known as distractors) from reading comprehension MCQ. The generated distractors should be in the context of the question but should not be semantically equivalent to the answer. Formally, *for given passage, question and answer triplet, generate a long, coherent, and grammatically correct wrong option*. Considering the fact that for a given triplet there can be many incorrect options that are completely different from each other, the problem is even more challenging. We use the English DG dataset from Maurya et al. [MD20b] which consists of approx 135k/17k/17k (train/validation/test) split. We were unable to find a suitable small training and test dataset in the Japanese language. For the Hindi language, we created a dataset called ***HiDG***[12] of 1k/1k/5k split. Similar to QG, to create input for ZmBART we concatenate the answer, question and passage in the same order and separate them with a special token <s>.

To create HiDG, we first extracted <passage, question, answer> triplets from English SQuAD 1.1 with at least a total of 150 tokens in the triplet. We generate distractors for these examples using the model proposed by Maurya et al. [MD20b]. The distractors were translated to Hindi using Google Translator service. The translated distractors were manually verified or corrected (if necessary) by human annotators.

The evaluation of the task is challenging because (1) There can be more than one correct distractor. Automated evaluation metrics may not be able to capture this aspect as only one ground truth distractor is available and (2) It may be possible that the generated distractor is semantically similar to the answer with high lexical overlap with reference distractor in those situations lexical match-based metrics are not suitable. To evaluate the DG task we mainly rely on BERTScore and manual evaluation. Towards this effort, we consider a higher number of DG samples for manual evaluation. Automated and human evaluation scores indicate the superiority of ZmBART over the baseline models for this task.

5.5.5 Overall Results

Here, we will discuss overall performance and major findings:

Performance Comparison with Baselines: The proposed ZmBART model consistently outperforms all the baseline models across both languages, all tasks, and all evaluation metrics in the zero-shot setting. The only exception is human evaluation for the QG task, where the proposed model has competitive performance with baselines possible due to code-mixed 'wh-words' generation, as discussed in Section 5.5.2. In the rest of the setup, the ZmBART successfully mitigates the

[12] Implementation, dataset, pre-trained checkpoints and ZmBART generated text are available at https://github.com/kaushal0494/ZmBART

CF/AT problem and enables well-formed zero-shot text generation.

Performance of Baseline Models: The auxiliary task-based baseline models, i.e., MonoMask and ParaMask, outperform the MTPipeline-based model. This indicates the importance of auxiliary tasks. Furthermore, MonoMask performs better than ParaMask, showing the effectiveness of auxiliary tasks with monolingual data. The proposed ZmBART model outperforms all the baseline models and emerges as a state-of-the-art model.

Automated vs. Human Evaluation: All the automated evaluation metrics show similar trends across the tasks. A similar observation holds for human evaluation metrics. Moreover, the evaluation with different annotator sets ensures inter-annotator agreement, except for the QG task. The automated and human evaluation scores correlate with each other. The proposed ZmBART model outperformed all the baseline models across both types of evaluations.

Zero-shot vs. Few-shot Performance: The fine-tuning of the ZmBART model with an additional 1000 examples in a few-shot setting further boosts the model's performance. It can be observed that the performance gain is high as compared to the zero-shot setting. This indicates the adaptability of the ZmBART model with a small task-specific supervised dataset.

Performance for Hindi vs. Japanese Language: Although the performance comparison across languages is generally not possible due to different tokenization schemes (affecting lexical overlap metrics), differences in language representation (affecting BERTScore metrics), and subjective bias are introduced by human evaluators (especially when comparing Hindi and Japanese evaluators). However, the above acceptance score for both languages indicates the ZmBART performs reasonably well for both languages.

Performance across Different Tasks: Except for human evaluation performance for the QG task, the proposed model consistently outperformed all tasks. It can be observed that the zero-shot performance (both human and automated) of DG is reasonably high, indicating ZmBART's intelligence in handling challenging tasks like DG. Furthermore, we have not modified a single parameter of the model across tasks, indicating the scalability of the proposed model.

Generalization of ZmBART Model: With these results, we now want to understand *whether the ZmBART is able to generalize across multiple tasks or favors specific tasks by considering the spurious correlation between auxiliary task and downstream NLG tasks objectives.* Among the tasks considered in this work, we see that the generation of meaningful summaries/headlines requires understanding/abstracting of input text, which is unlikely to be obtained by repeating sentences from input passages, as done in the auxiliary task. ZmBART achieves good zero-shot and few-shot on ATS and NHG over strong baselines. The generated headlines and summaries were found to be mostly abstractive; they do not contain large continuous sequences from input text. As described in Sections 5.5.2 and 5.5.4, Question Generation and Distractor Generation are more challenging tasks and have objectives vastly different from the auxiliary task's objective. Even for these tasks, decent evaluation scores and improvements over the baselines across the considered LRL indicate that the solutions are not spurious. The incorporation of auxiliary tasks improves the performance of diverse downstream tasks on real benchmark datasets and does not favor any specific task or dataset.

To summarize, we have performed experiments for 14 different task-setup combinations involving two LRLs. With four tasks in Hindi and three tasks in Japanese, and each task in zero-shot and few-shot setup, we provide a detailed comparative evaluation for the tasks. The tasks are of different natures, and each task offers its own unique challenge. We critically analyze the performances to show the robustness and the range of applicability for the proposed ZmBART framework.

5.6 Further Analyses and Discussions

In this section, we provide further analyses, ablation, and experiments to understand the impact and effect of different modeling components of ZmBART. This will also highlight the reasoning for the different design choices.

5.6.1 ZmBART Performance for HRL

Table 5.6 presents automated evaluation results of the ZmBART model for high-resource English. This has been presented in two setups (with and without the

Task	Setup	BL	R-1	R-2	R-L	BS
NHG	without Auxiliary Task	15.9	43.15	21.25	40.77	90.13
	with Auxiliary Task	15.9	43.22	21.33	40.88	90.13
QG	without Auxiliary Task	21.4	52.66	26.63	51.25	92.41
	with Auxiliary Task	20.6	53.20	26.53	51.37	92.18
ATS	without Auxiliary Task	15.8	39.52	18.00	37.91	90.10
	with Auxiliary Task	16.0	40.01	18.11	38.29	90.20
DG	without Auxiliary Task	10.0	31.87	14.59	31.30	89.42
	with Auxiliary Task	10.3	31.76	14.89	31.18	89.33

Table 5.6: ZmBART model performance for high-resource English language in two setups: without and with the proposed auxiliary task. Results are reported across all four tasks with five automated evaluation metrics.

proposed auxiliary task) across all four downstream NLG tasks. The evaluation is done with English test data in a supervised setting. With this experiment, we aim to (1) assess the performance of ZmBART for HRL and (2) understand the effect of auxiliary tasks on ZmBART in the context of HRL. We observe that the performance of HRL is reasonably high, and there is no significant performance degradation of ZmBART due to the inclusion of adaptive pre-training steps for HRL. Even the auxiliary task helps achieve a slight improvement. This concludes that ZmBART can be adapted as a replacement for the original mBART model, even for HRL and this single model is effective for both HRL and LRLs.

5.6.2 Effect of Auxiliary Task for LRLs

Table 5.7 presents the zero-shot results of ZmBART for ATS and QG tasks in two setups: with and without auxiliary task adaptive training. It can be observed that, without the auxiliary task, lexical match-based scores are poor because the decoder generates code-mixed outputs due to the CF/AT problem. BERTScore still remains reasonable without auxiliary tasks, indicating that even the code-mixed generations are semantically relevant to the reference. However, well-formed generation in the target LRL is enabled only after the inclusion of the auxiliary task. The auxiliary task contributes in two ways: it enables zero-shot well-formed generation and improves cross-lingual transfer from HRL to LRLs. We have similar observations for NHG and DG tasks.

Experiment Setup	**Abstractive TS**			**Question Generation**		
	R-1	**R-2**	**R-3**	**BL**	**R-L**	**BS**
Hindi Language						
without Auxiliary Task	4.34	0.10	3.19	0.9	16.64	70.72
with Auxiliary Task	21.27	5.30	17.64	4.4	26.51	74.19
Japanese Language						
without Auxiliary Task	6.80	0.11	5.30	6.7	33.07	70.35
with Auxiliary Task	36.60	15.26	29.89	18.8	38.74	75.91

Table 5.7: Zero-shot performance of ZmBART with and without auxiliary task for Hindi and Japanese languages

Approach	**Setup**	**BL(hi/ja)**	**R-L(hi/ja)**	**BS(hi/ja)**
Freezing Components	Freeze word embedding (WE)	2.5/13.6	21.55/31.99	72.02/73.18
	Freeze WE + Subset of Encoder & Decoder layers	2.9/15.3	22.62/36.60	72.24/72.98
	Freeze WE + Encoder layers	2.2/13.8	19.69/36.91	69.73/72.97
	Freeze WE + Decoder layers (ZmBART)	**4.4/18.8**	**26.51/38.74**	**74.19/75.91**
Parameter Regularization	Elastic Weight Consolidation (EWC)	2.1/11.6	18.21/29.47	68.36/72.91

Table 5.8: Different approaches to mitigate the catastrophic forgetting problem for QG task. hi: Hindi, ja: Japanese

5.6.3 Ablation Study for Catastrophic Forgetting Problem

We have experimented with two trending approaches to mitigate the issue of catastrophic forgetting, drawing inspiration from continual learning approaches [VdVT19]. These methodologies include (a) Freezing parameters of the model components and (b) Parameter Regularization. Tables 5.8 and 5.9 present the zero-shot automated evaluation results with a different combination of two approaches for QG and NHG tasks, respectively. Notably, our proposed modeling setup (i.e., ZmBART) demonstrates the best performance. Similar trends have been observed in the case of ATS and DG tasks.

Approach	**Setup**	**R-1(hi/ja)**	**R-2(hi/ja)**	**R-L(hi/ja)**
Freezing Components	Freeze word embedding (WE)	13.02/26.07	05.67/03.96	12.45/17.62
	Freeze WE + Subset of Encoder & Decoder layers	14.27/25.72	06.70/03.21	13.76/18.28
	Freeze WE + Encoder layers	09.81/22.67	04.10/02.38	09.66/13.68
	Freeze WE + Decoder layers (ZmBART)	**34.94/35.25**	**19.38/09.24**	**32.74/19.92**
Parameter Regularization	Elastic Weight Consolidation (EWC)	12.01/22.16	05.43/03.11	11.22/16.31

Table 5.9: Different approaches to mitigate the catastrophic forgetting problem for NHG task. hi: Hindi, ja: Japanese

Setup	News Headline Generation			Question Generation		
	R-1	**R-2**	**R-3**	**BL**	**R-L**	**BS**
mBART with WD	50.61	34.32	49.01	6.1	31.20	77.01
mBART	51.49	35.04	49.64	7.1	32.96	77.61
ZmBART	51.81	35.04	50.07	6.9	32.82	77.40
ZmBART without WD	**52.37**	**35.52**	**50.50**	**7.9**	**34.49**	**78.39**

Table 5.10: Few-shot results with different architectural setups for the Hindi language. WD: word embeddings and all decoder parameters are frozen

5.6.4 Effect of Architecture on Few-shot Training

In this setup, we experiment with few-shot training with mBART (directly fine-tuned on task-specific supervised English data) and ZmBART (trained with auxiliary task and fine-tuned with English data). The results are presented in Table 5.10. We find that ZmBART does better than mBART in corresponding setups. Moreover, although freezing the decoder layer and word embeddings helps in a zero-shot setting, it is natural and useful to unfreeze them during few-shot training.

5.6.5 Few-shot Performance vs. Supervised Data Size

Fig. 5.3 shows the trends of few-shot training of ZmBART with respect to supervised Hindi and Japanese training data for ATS and QG tasks, respectively. We observe that even with a small number of supervised examples (e.g., 100), the model achieves a reasonable few-shot performance. The improvement tends to be minimal after training with 1000 examples. We found that the different tasks exhibit similar trends.

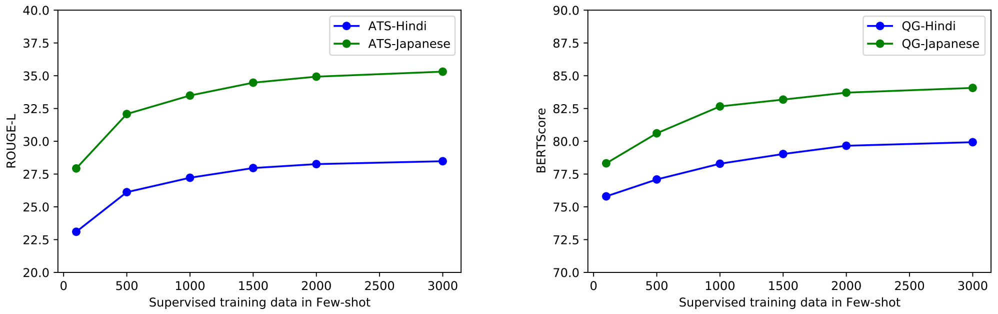


Figure 5.3: Trends of few-shot performance of ZmBART with supervised Hindi and Japanese data for ATS (left) and QG (right) tasks.

5.7 Conclusion

In this paper, we propose a novel unsupervised framework, ZmBART, to address the issue of catastrophic forgetting and enhance cross-lingual transfer. The framework transfers supervision from HRL to LRLs, enabling well-formed zero-shot generation in LRLs without the need for parallel or pseudo-parallel/back-translated data. ZmBART is directly applied to multiple downstream NLG tasks and LRLs without any modification of hyper-parameters. The model incorporates a carefully designed auxiliary task to improve the multilingual embedding space and facilitates a *warm-up* start for well-formed zero-shot generation. We conducted experiments in three languages across 18 task setups: four supervised tasks in English, four tasks in Hindi (each with zero-shot and few-shot settings), and three tasks in Japanese (each with zero-shot and few-shot approaches). With the exception of zero-shot question generation tasks, for all other tasks involving LRLs, the proposed model consistently generated high-quality results, as validated by automated and manual evaluation.

5.8 Insights, Limitations and Future Work

Insights: As the auxiliary task is similar to NHG or ATS tasks, it may appear that the auxiliary task is biased towards these tasks, which leads to better performance. However, the model performs equally well for very different tasks like QG and DG tasks, which nullifies this assumption. We have not modified any single model hyper-parameter values for different tasks. We also experimented with different objectives for auxiliary tasks; however, the RAND-SUMMARY objective (as in ZmBART) performed the best. We explored multiple continual learning techniques to mitigate CF; however, freezing model components work best along with adaptive pre-training and language tag. We observed that several generated questions in zero-shot start with English *wh-words*, and the first word is code-mixed. This is possibly due to English interrogative sentences often introducing *wh-words* at the beginning, which may not be the case with Hindi and Japanese. However, the high BERTScore indicates semantic correctness. Furthermore, such code-mixing in human evaluation is somewhat acceptable with Hindi evaluators; however, it is not acceptable with Japanese evaluators, resulting in lower human evaluation scores for the QG task for the Japanese language. This is concurrent work with the adapter-based models [HGJ+19, PKR+21]. We have experimented with the mBART language model; however, the proposed model is language model agnostic. This has been validated in the next Chapter.

Limitations: One limitation of this work is that adapting to a new language may require retraining the mBART model with the new languages during the adaptive fine-tuning phase. This aligns with the language model pretraining, as including more languages may necessitate retraining. However, adapting to a new task does not require any retraining with the auxiliary task, and no hyperparameter modifications are needed.

Future Work: In the future, we aim to expand upon this work by incorporating more LRLs and tasks. We will also investigate alternative auxiliary tasks to improve cross-lingual transfer signals. Specifically, our objectives include adapting to new languages, either through re-training or building upon the ZmBART. Additionally, two crucial futures of exploration are (i) Enhancing cross-lingual transfer by leveraging explicit linguistic features from languages (Chapter 6) and (ii) Enabling language technologies for LRLs that lack parallel data, possess limited monolingual data, and are absent from large multilingual language models (Chapter 7).

Chapter 6

Meta-Learning Approach to Improve Zero-Shot Cross-Lingual Transfer and Generation

6.1 Introduction

Zero-shot modeling is a promising research direction to enabling language technology in low-resource languages (LRLs). However, for natural language generation (NLG) tasks, this modeling presents its own challenges, including catastrophic forgetting (CF)/ accidental translation (AT) problems, limited learning data, uneven supervision transfer and many more. The previous chapter deals with mitigating the CF/AT problem, and as a side effect, it improves cross-lingual transfer. In this chapter, we explicitly focus on improving cross-lingual transfer from high resource language (HRL) to LRLs by considering linguistic information such as language structure and typological features. We again focus on NLG tasks, zero-shot setting, and a large set of LRLs. Before delving into the details, let's take a step back and understand why this research has a direct social impact and its applicability to a wide audience.

There are more than 7000 known spoken languages across the globe. 95% of the world's population does not speak English as their first language and 75% does not speak English at all[1]. Most of the languages are LRLs as they do not have adequate resources for NLP research [JSB+20]. On the other hand, a vast majority of studies in NLP research are conducted on English data [Ben19]. To democratize the NLP research for the benefit of the large global community, it is essential to focus on

[1] https://www.ethnologue.com/statistics

the non-English languages. However, creating/collecting task-specific annotated data for all the languages is expensive and time-consuming. Moreover, human languages are dynamic as new words and domains are added continuously. An alternative solution is to explore NLP modeling techniques that enable training models with HRL, like English and transferring supervision to LRLs that have limited or no annotated data. Recently, there has been promising progress on cross-lingual transfer learning research [HRS+20, ARY20a], but supervision transfer from HRL to LRLs is *non-uniform*, which leads to large performance gaps. Such performance gaps are observed because the LRLs, which are less similar to HRL, have weak supervision transfer as models do not account for cultural and linguistic differences in the modeling [LOYS19, BAN21]. *This paper is a step towards bridging this gap via meta-learning and language clustering.*

Meta-learning or *learning to learn* [BBC90] is a learning paradigm where the model is trained on diverse tasks and quickly adapts to new tasks given a handful of examples. It has emerged as a promising technique in Machine Learning [FAL17, KZS+15], Natural Language Understanding [MHM21, YZJZ20] and Machine Translation [GWC+18]. This work - to the best of our knowledge - is the first attempt to study *meta-learning techniques for cross-lingual natural language generation (X_{NLG})*. Particularly, we focus on zero-shot $\mathrm{X}_{\mathrm{NLG}}$ for LRLs. Unlike NLU tasks, we observe that zero-shot NLG is a more challenging setup as the text should be generated in unseen languages (which suffers from CF/AT problem) and is expected to be grammatically coherent, semantically correct, and fluent. Moreover, the supervision transfer is often non-uniform.

Considering these concerns, we propose a novel modeling framework called Meta-$\mathrm{X}_{\mathrm{NLG}}$ [MD22] for cross-lingual transfer and generation with language cluster and meta-learning. First, we cluster a large set of LRLs into different clusters and obtain the centroid and non-centroid languages for each cluster. Then, a meta-learning algorithm is trained with centroid languages and evaluated with non-centroid LRLs in a zero-shot setting. This modeling effectively mitigates the uneven supervision transfer and boosts the performance of LRLs that are less similar to HRLs. With this work, we aim to address the following research problem: *Does meta-learning algorithm trained on typologically diverse languages (as training task) provide language-agnostic initialization for the zero-shot cross-lingual generation?*

Our main contributions with Meta-$\mathrm{X}_{\mathrm{NLG}}$ are listed below:

- We propose Meta-X_{NLG}[2] [MD22], a framework for effective cross-lingual transfer and generation based on the *Model-Agnostic Meta-Learning (MAML)* and *language clustering*. We have utilized the findings from ZmBART.
- We use language clustering to identify a set of meta-training languages. Training with these languages provides a more uniform cross-lingual transfer to less similar LRLs (with HRLs) in a zero-shot setting.
- We test Meta-X_{NLG} on two NLG tasks (Abstractive Text Summarization and Question Generation), using five popular datasets (XL-Sum, Wikilingua, MLQA, TyDiQA, and XQuAD), across 30 languages. We observe consistent improvement over strong baselines, including mT5.

6.2 Related Work

In this section, we will discuss two threads of existing literature: (1) cross-lingual generation and (2) meta-learning for NLP.

6.2.1 Cross-Lingual Generation

Traditional approaches for cross-lingual generation use machine translation (MT) in the modeling pipeline [WLX10b, ASC+18, DYZ+19]. MT-based approaches have inherent problems like scalability, and translations are error-prone. In the case of LRLs, these errors are more pronounced, hindering usability. Recently, cross-lingual transfer approaches are gaining attention. These methods use parallel data [CDW+20b] and small annotated datasets [KJM+19b] in the modeling. Lewis et al. [LGG+20a] fine-tune a pre-trained model with multiple LRLs and evaluate a single target language in a zero-shot setting. In the same line of research, we have proposed an unsupervised approach [MDKD21a] to mitigate CF/AT problem and enable cross-lingual transfer. It has been observed that such cross-lingual transfers are not uniform across the languages [LCL+19, BAN21]; supervision transfer is ineffective if the LRLs are less similar to HRL in a zero-shot setting. Unlike these, we propose a meta-learning approach to enable a more uniform and effective cross-lingual transfer in a zero-shot setting.

[2] Code & pre-trained models: https://github.com/kaushal0494/Meta_XNLG

6.2.2 Meta-Learning for NLP

Recently, meta-learning has been actively applied for many NLP applications [BJMM20, GHZ+19] including text classification [vdHYMS21], NER [WLW+20], task-oriented dialogue and QA [MKD+21] and many more. Tarunesh et al. [TKK+21] propose joint meta-learning approach on multiple languages and tasks from XTREME benchmark [HRS+20]. Close to our work, [NBBA20] propose a meta-learning approach for cross-lingual transfer on NLI and QA, both NLU tasks. The authors use one or two randomly selected languages for meta-training. In contrast, we provide a systematic approach based on language clustering to identify the right meta-training languages. Moreover, to the best of our knowledge, ours is the first effort that employs meta-learning for downstream NLG tasks to improve cross-lingual transfer and generation.

6.3 Background: Meta-Learning (MAML)

Meta-learning algorithms aim to learn common (meta) structures among multiple tasks such that the new tasks are adapted quickly given few training instances. It is also known as *few-shot learners.* Among several meta-learning algorithms, we focus on optimization-based algorithms, i.e., Model Agnostic Meta-Learning (MAML) [FAL17] due to its recent success in multiple NLP and computer vision tasks. MAML progresses in two phases: *meta-training* and *adaptation*. In the meta-training phase, the model learns a good initialization of parameter values by repeatedly simulating the learning process with multiple training tasks. In the adaptation phase, these learned parameters are quickly adapted to new tasks. The underlying constraint is that *all tasks should share some common structure (or come from a task distribution).* The world's different languages follow this constraint as they come into existence with a common goal of communication and share some structure. For meta-learning purposes, we treat each language as a task.

Unlike traditional machine learning, meta-learning has *meta-train* and *meta-test* data splits for meta-training and meta-adaptation (aka. adaptation), respectively. Each split consists of tasks that are sampled from a distribution $p(\mathcal{D})$ over task datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ where $\mathcal{D}_i$ is associated with i^{th} task $\mathcal{T}_i$. Each $\mathcal{D}_i$ has *support set* and *query set* $\mathcal{D}_i = \{\mathcal{S}_i, \mathcal{Q}_i\}$. *Support set* and *Query set* are analogous to train and test splits of traditional machine learning. We use f_θ to denote a neural network model parameterized by θ.

Meta-training has two levels of optimization: *inner-loop optimization* and *outer-loop optimization*. In the inner-loop optimization, for each sampled task $\mathcal{T}_i$, the task-specific model parameters θ_i^m are updated by m iterations of stochastic gradient descent (SGD) with support set $\mathcal{S}_i$. The overall model parameters θ are learned to optimize the performance of models $f_{\theta_i^{(m)}}$ on query sets $\mathcal{Q}_i$ across datasets $p(\mathcal{D})$ in the outer-loop optimization. The MAML [FAL17] objective is:

$$\theta^* = \arg\min_{\theta} \sum_{D_i \sim p(D)} \mathcal{L}_i(f_{\theta_i^{(m)}}) \tag{6.1}$$

where $\mathcal{L}_i(f_{\theta_i^{(m)}})$ is the loss obtained on query set for task $\mathcal{T}_i$ and $f_{\theta_i^{(m)}}$ is obtained after m iteration of SGD update with Task $\mathcal{T}_i$ as:

$$f_{\theta_i^{(m)}} = f_\theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta)$$

In outer-loop optimization, MAML performs *MetaUpdate* which a batch as:

$$\theta = \theta - \beta \nabla_\theta \sum_{D_i \sim p(D)} \mathcal{L}_i(f_{\theta_i^{(m)}}) \tag{6.2}$$

Where α is the inner-loop learning rate and β is the meta (outer-loop) learning rate. In the adaptation phase, the model is initialized with learned optimal meta-parameters θ^*, which is updated by a few steps of SGD with a support set of the meta-test dataset (aka. few-shot learning) and directly evaluated on the query set of the meta-test dataset. Our aim is to perform zero-shot evaluation, so we skip the adaptation phase and directly evaluate the learned model θ^* on meta-test datasets. In our case, we do not have support and query split for meta-test datasets hence, our proposed model is evaluated with all examples of meta-test datasets.

6.4 Methodology

In the proposed Meta-X$_{\text{NLG}}$ framework, we first cluster the available languages and identify the centroid languages. Then, we train a model with MAML on centroid languages to obtain an optimal initialization of parameters. Finally, the learned model is deployed to generate text in the zero-shot setting. Figure-6.1 provides an overview of the proposed framework. We now deep dive into the details of each component of the Meta-X$_{\text{NLG}}$ framework.

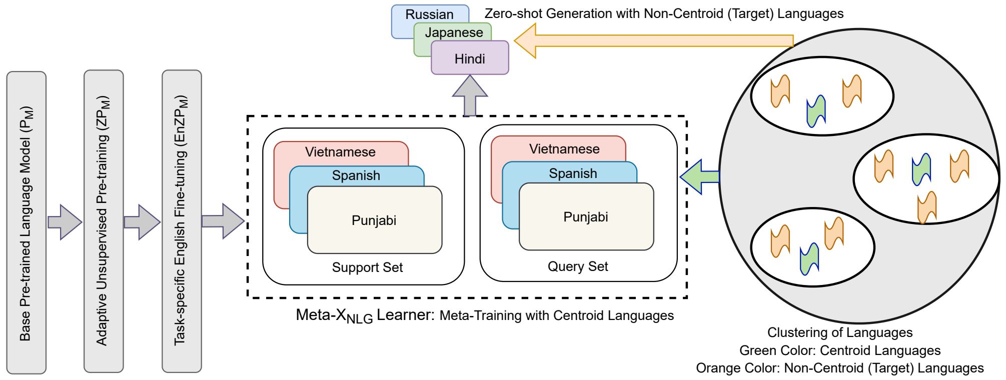

Russian
Japanese
Hindi
Zero-shot Generation with Non-Centroid (Target) Languages
Base Pre-trained Language Model (P_M)
Adaptive Unsupervised Pre-training (ZP_M)
Task-specific English Fine-tuning ($EnZP_M$)
Vietnamese
Spanish
Punjabi
Support Set
Vietnamese
Spanish
Punjabi
Query Set
Meta-X_{NLG} Learner: Meta-Training with Centroid Languages
Clustering of Languages
Green Color: Centroid Languages
Orange Color: Non-Centroid (Target) Languages

Figure 6.1: An overview of Meta-X_{XNLG} framework

6.4.1 Language Clustering

The languages can be clustered in two ways: (1) by considering language family and (2) by exploiting similarities among learned language representations. Further, learned language representations are obtained using typological features [LML+17] from linguistic knowledge bases like WALS [DH13] and Glottolog [HFH17] or learned language tag representations from tasks like machine translation [MNL17]. Recently, Oncevay et al. [OHB20] fuse typologically learned and task-learned language representations using singular vector canonical correlation (SVCC) analysis to obtain multi-view language representation. Further, the authors cluster languages using these rich multi-view language representations through hierarchical clustering. We utilize this existing language cluster approach and multi-view language representations in our proposed framework.

We have considered 30 languages. Most of the languages do not have training resources for downstream NLG tasks and are considered LRLs. We first cluster the considered languages using the approach proposed by Oncevay et al. [OHB20]. A sample hierarchical clustering is illustrated as a dendrogram in Fig. 6.2 and final clustering is shown in Table 6.1. Next, we aim to identify a representative language (*centroid language*) for each cluster. Formally, given a cluster $C = \{L_1, L_2, \dots L_t\}$, where each L_i is multi-view representation of i^{th} language, the centroid language $L^* \in C$ is defined as:

$$L^* = \arg\min_{L_i \in C} \sum_{L_j \in C} d(L_j, L_i). \tag{6.3}$$

Here d is the cosine distance. Intuitively, we have incorporated the typological feature-based learned language representation for systematic clustering. This indirectly results in linguistically inspired clustering, where the languages with similar

Cluster-1(14)	Cluster-2(8)	Cluster-3(8)
hi,ur,te,tr,ja,fi,ko,gu, bn,mr,np,ta,pa,sw	es,it,pt,ro, nl,de,en,fr	ru,cs,vi,th, zh,id,el,ar

Table 6.1: Clustering of considered 30 Languages

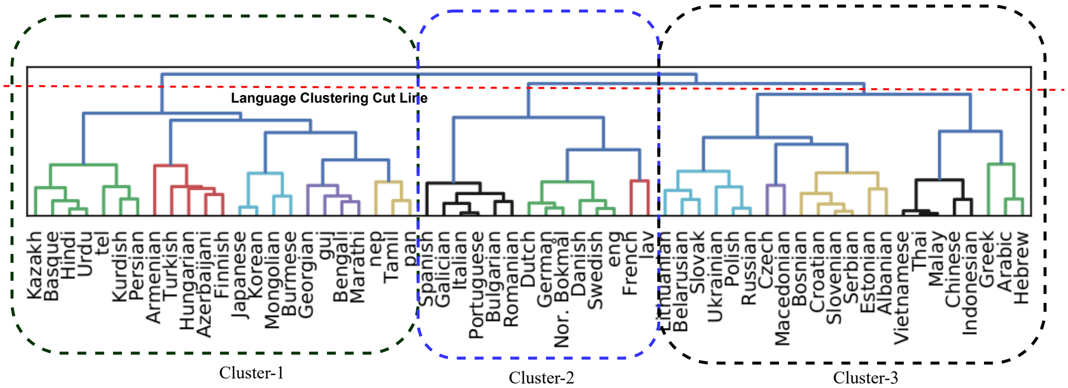


Figure 6.2: Language clustering is based on a multi-view representation [OHB20]. We intentionally showcase more than 30 languages in the clustering, which will prove useful for scaling the proposed work in the future. The dotted red line represents the cut-line used to obtain different sets of clusters. Here, we show three clusters that perform best in our case.

features group into the same cluster. This is an important property the Meta-X$_{\text{NLG}}$ required for generalization (more details on this later).

6.4.2 Meta-X$_{\text{NLG}}$ Training

The Meta-X$_{\text{NLG}}$ framework consists of five training steps: selection of base pre-trained model, adaptive unsupervised pre-training, fine-tuning with HRL, meta-training with LRLs, and meta-adaptation for zero-shot. The motivation and details of each step are provided below:

1. **Selection of Base Pre-trained Model (P_M):** Our approach is model-agnostic, therefore any state-of-the-art sequence-to-sequence multilingual pre-trained language model (P_M; like mBART, mT5, etc.) can be used. We selected mT5 due to its superiority on many NLP tasks [XCR$^+$21] and large LRL coverage.

2. **Adaptive Unsupervised Pre-training (ZP_M):** Zero-shot cross-lingual generation often suffers from catastrophic-forgetting/accidental translation

[XCR+21] and other generation problems. To overcome these issues, we utilized findings from the ZmBART model to develop ZP_M (or ZmT5). We will provide more details on this in Section-6.4.3.

3. **Fine-tuning ZP_M with HRL (i.e., English):** It is often observed that downstream LRLs applications benefit when supervision is transferred from HRL [HRS+20]. Following the trend, we fine-tune the ZP_M model with the large task-specific English supervised data and call this model $EnZP_M$ (EnZT5) with parameters θ_p.

4. **Meta-Training with Low-resource Centroid Languages:** We use a small validation dataset of each centroid language as the *meta-train* dataset. First, the meta-learner is initialized with the $EnZP_M$ parameters. Then, a batch D_i from a centroid language (T_i; aka. tasks) validation dataset is randomly sampled. Further, D_i is equally split into support set S_i and query set Q_i such that they are mutually exclusive. m-step gradient update is done in the inner loop using S_i. This is repeated for all the centroid languages (i.e., training tasks). Finally Meta-Update is done using mean loss computed on Q_i as shown in Equation 2. This is repeated for all the tasks/languages over multiple batches. The batches are sampled uniformly from all centroid languages. The formal description is shown in Algorithm-1.

5. **Meta-adaptation for Zero-shot Evaluation:** The meta-learned model f_{θ^*} from the previous step can be directly evaluated on the test sets of the target languages (non-centroid LRLs) in zero-shot setting. The proposed framework can be easily extended to a few-shot setting. In this setting, the meta-learned model can be fine-tuned on a small number of validation set examples with standard supervised learning and evaluated on the test sets of target languages. In this work, we only focused on zero-shot setting.

In the proposed Meta-X$_{\text{NLG}}$ algorithm, the centroid LRLs act as *Meta-Training* tasks/languages, and the rest of the non-centroid LRLs as *Target* (aka. evaluation) tasks/languages. In this setup, the best performing model should hold two properties, i.e., *Intra-cluster Generalization* and *Inter-cluster Generalization.* In the proposed framework, training with a single centroid language leads to better cross-lingual transfer capability within the cluster, and using multiple centroid languages extends the transfer capability to multiple closely-knit clusters and increases coverage. In this way, *Intra-cluster Generalization* and *Inter-cluster Generalization* are

Algorithm 1 Meta-X$_{\text{NLG}}$: Meta-Learning Algorithm for Cross-lingual Generation

```
Require: Task set distribution $p(D)$; pre-trained model $EnZP_M$ (P) with parameters $\theta_P$;
    meta-learner $f_\theta$ with parameter $\theta$; number of centroid languages $c$
Require: $\alpha$, $\beta$: step size hyper-parameters
Initialize $\theta \leftarrow \theta_P$
while not done do
    for each $T_i$ in $T$ $T = T_1, T_2, \ldots T_c \sim p(D)$ from centroid languages do
        Initialize $\theta_i \leftarrow \theta$
        Sample a batch $D_i$ using the development set of task $T_i$
        Split $D_i$ to form support set $S_i$ and query set $Q_i$
        for all inner_iter steps $m$ do
            Compute $\nabla_{\theta_i^{(m)}} L_{T_i}^{S_i}(P_{\theta_i^{(m)}})$
            Do SGD: $\theta_i^{m+1} = \theta_i^m - \alpha \nabla_{\theta_i^{(m)}} L_{T_i}^{S_i}(P_{\theta_i^{(m)}})$
        end for
        MetaUpdate: $\theta = \theta - \beta \nabla_\theta \sum_{j=1}^{b} L_{T_i}^{Q_i}(P_{\theta_i^{(m)}})$
    end for
end while
Do zero-shot/few-shot learning with meta-learner $f_{\theta^*}$ where $\theta^*$ is learned optimal parameters of meta-learner.
```

achieved. However, there is a trade-off between the number of clusters (the number of meta-training languages) and generalization. If there is a single cluster (a single meta-training language), then the model tries to over-generalize for different typological structures and fails in the attempt. On the other extreme, if there are too many centroid languages (many typologically diverse structures in meta-training), then the learning possibly gets distracted. In both cases, the model will be unable to learn a reasonable structure (the required generalization) and perform poorly. Section-6.6.2 consists of discussions and empirical evidence. Our experiments suggest that three clusters across considered languages provide the best performance. These three clusters are always fixed irrespective of the datasets and underlying tasks. The composition of the clusters (with three clusters) is shown in Table-6.1. See Figure-6.2 for more details on the clustering.

6.4.3 Avoiding Catastrophic-Forgetting (CF)/Accidental Translation (AT) Problem:

It has been observed that popular pre-trained models in well-formed generations for unseen low-resource (zero-shot) languages. Broadly, they suffer from Accidental Translation (AT), where the model generates the whole/part of the output in the fine-tuning language [XCR+21]. This happens when the model forgets the learning obtained before fine-tuning. This is analogous to the Catastrophic-Forgetting problem [CDW+20b] in multi-task setup, where the model forgets the learning about the previous task. For language generation, this also leads to problems like improper predictions, structural and normalization errors, etc., as the different languages differ in morphology, phonology, subject-verb-object ordering, etc. To mitigate/reduce these problems, we utilize the findings from ZmBART [MDKD21a], and propose the following solution in Meta-X_{NLG} framework.

- **Adding Language Tag:** We concatenate *<fxx> <2xx>* where *xx* is language code as per ISO 693-2 standard.
- **Adaptive Unsupervised Pre-training**: Further train the base pre-trained model P_M with small monolingual data from all considered languages and Rand-Summary objective.
- **Freezing model Components :** One of the key approaches to mitigate the CF problem is freezing model parameters. Inspired by ZmBART, freezing all token embeddings and the decoder parameters of the model works best. We adapted these findings to HRL fine-tuning and meta-training steps.

We observed that the above settings work better to mitigate (or reduce) the CF/AT problems.

6.5 Experiment Setup

We investigate Meta-X_{NLG}'s performance on two downstream NLG tasks, five public datasets, and 30 languages. mT5 pre-trained model [XCR+21] is used as the base model. The model performance is compared with two strong baselines in a zero-shot setting.

6.5.1 Tasks and Datasets

Abstractive Text Summarization (ATS):

ATS is the task of *generating grammatically coherent, semantically correct, and abstractive summary given an input document.* We use two publicly available datasets: XL-Sum [HBI+21] and Wikilingua [LDCM20a].

- **XL-Sum** is a large comprehensive dataset where article-summary pairs are extracted from BBC and annotated by professional annotators. It covers 44 languages including very low-resource languages like Nepali and Swahili. Due to computational limitations, we consider only 23 languages.

- **Wikilingua** is a large-scale dataset covering 18 languages. Article and summary pairs are extracted from WikiHow[3]. It is *how-to guides* on diverse topics written by human annotators for software and tools. We consider all 18 languages in our experiments.

6.5.2 Question Generation (QG):

In QG, *given an input passage and an answer, it aims to generate semantically and syntactically correct questions that can produce the answer.* We use three publicly available multilingual question and answering (QA) datasets: MLQA [LOR+20b], TyDiQA [CCC+20b], and XQuAD [ARY20a]. Each instance is a triplet of <passage, question, answer>. We concatenated *answer* and *passage* with delimiter *<s>* in the same order as input for models.

- **MLQA** is a multi-way parallel extractive QA evaluation dataset available in 7 languages. Authors automatically extracted paragraphs from Wikipedia articles in multiple languages that have the same or similar meanings. Authors crowd-sourced questions in English and translated them into target languages by professional translators. As our framework is based on supervision transfer, we only consider the evaluation data instance where input and target text languages are the same. In this way we have seven datasets for seven languages.

- **XQuAD** dataset is translated from the development set of SQuAD v1.1 [RZLL16c] by professional human translators into 10 languages. Each language has 1190 question-answer pairs. SQuAD is a popular question-answering dataset consisting of around 100k <passage, question, answer> triplets. We added an additional Japanese language data set [TSKK19b] which is created with similar goals and has the same format.

- **TyDiQA** is another QA dataset with 204K question-answer pairs in 11 typologically diverse languages. Unlike MLQA and XQuAD, it is directly collected in each language and does not involve any translation. We use, *TyDiQA-GoldP*

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

datasets which are guaranteed to have extractive nature. We added Tamil as an additional language that shares the same format and is created with similar goals.

We use English data from XL-Sum and Wikilingua for the English fine-tuning step while experimenting with the respective dataset. MLQA, TyDiQA and XQuAD do not have any English training data. Following the trend [LOR+20b, CCC+20b] we use SQuAD v1.1 training data at the English fine-tuning step. The validation dataset is used for meta-training (only for selected centroid languages) and test dataset is used for the zero-shot generation. The summary of all languages and data statistics are presented in Table 6.2.

For each dataset, we grouped the languages into three fixed clusters as per Table-6.1 and found the centroid language as described in Section-6.4.1. English is the high resource language and is only used for supervised fine-tuning as described in section-6.4.2 so it will not be part of any cluster. To make it more concrete, the XQuAD dataset has 11 low-resource languages (excluding English), the centroid (Meta-training) languages are <tr, es, th> and non-centroid (Target) languages are <hi, to, de, ar, vi, zh, ru, el>. For each task and dataset, Table 6.3 summarizes the clustering, centroid, and non-centroid languages.

6.5.3 Baselines

Due to the unavailability of prior zero-shot models for considered datasets, we design strong baselines based on recent cross-lingual/multilingual models and architectures.

- **EnZmT5:** Inspired by Maurya et al. [MDKD21a], we further train the mT5 model with a small monolingual dataset from 30 languages, using an auxiliary task training objective, followed by task-specific English fine-tuning (similar to the first three steps of the Meta-X_{NLG} model proposed in Section 6.4.2). Finally, it is then directly evaluated in a zero-shot setting on the target LRLs.

- **FTZmT5:** In this model we fine-tune EnZmT5 baseline on all centroid languages. This will ascertain that the improvement of Meta-X_{NLG} is not due to simply training on more datasets in different languages. This is close to the Lewis et al. [LGG+20a]'s model, but they use different datasets.

While training EnZmT5 and FTZmT5, we use all applicable precautions as suggested in sections-6.4.3 and grid search to find the best hyper-parameters.

SN	**Language**	**ISO-2**	**ISO-3**	**Adap. PT** **train/valid/test**	**XL-Sum** **test**	**Wikilingua** **test**	**MLQA** **test**	**TyDiQA** **test**	**XQuAD***** **test**
1	English*	en	eng	5k/1k/1k	300k/11k/11k	100k/13k/28k	90k/10k/11k	90k/10k/11k	90k/10k/11k
2	Hindi	hi	hin	5k/1k/1k	8847	1983	4918	-	1190
3	Urdu	ur	urd	5k/1k/1k	8458	-	-	-	-
4	Telugu	te	tel	5k/1k/1k	1302	899	-	5563	-
5	Turkish	tr	tru	5k/1k/1k	3397	-	-	-	1190
6	Finnish	fi	fin	5k/1k/1k	-	-	-	6855	-
7	Japanese	ja	jpn	5k/1k/1k	889	2529	5000**	-	-
8	Korean	ko	kor	5k/1k/1k	550	2435	-	1620	-
9	Gujarati	gu	guj	5k/1k/1k	1139	-	-	-	-
10	Bengali	bn	ben	5k/1k/1k	1012	-	-	2390	-
11	Marathi	mr	mar	5k/1k/1k	1362	-	-	-	-
12	Nepali	np	nep	5k/1k/1k	725	-	-	-	-
13	Tamil	ta	tam	5k/1k/1k	2027	-	-	368**	-
14	Punjabi	pa	pan	5k/1k/1k	1026	-	-	-	-
15	Swahili	sw	swa	5k/1k/1k	987	-	-	2755	-
16	Spanish	es	spa	5k/1k/1k	4763	22626	5253	-	1190
17	Italian	it	ita	5k/1k/1k	-	10187	-	-	-
18	Portuguese	pt	por	5k/1k/1k	7175	16326	-	-	-
19	Romanian	ro	ron	5k/1k/1k	-	-	-	-	1190 -
20	Dutch	nl	nld	5k/1k/1k	-	6248	-	-	-
21	German	de	deu	5k/1k/1k	-	11667	4517	-	1190
22	French	fr	fra	5k/1k/1k	1086	12728	-	-	-
23	Russian	ru	rus	5k/1k/1k	7780	10577	-	6490	1190
24	Czech	cs	ces	5k/1k/1k	-	1438	-	-	-
25	Vietnamese	vi	vie	5k/1k/1k	4013	3916	5459	-	1190
26	Thai	th	tha	5k/1k/1k	826	2949	-	-	1190
27	Chinese (Sim)	zh	zho	5k/1k/1k	4670	3772	5137	-	1190
28	Indonesian	id	ind	5k/1k/1k	4780	9495	-	5702	-
29	Greek	el	ell	5k/1k/1k	-	-	-	-	1190
30	Arabic	ar	ara	5k/1k/1k	4689	5840	5335	14805	1190

Table 6.2: Details of the datasets used in Meta-X$_{\text{NLG}}$. For adaptive pre-training, a small 5k/1k/1k dataset is used. Ada.PT: Adaptive unsupervised pre-training. *-English is a high-resource language for which all three splits were used, as shown in Row 1. **-Additional language added to the dataset. ***-The dataset does not have a validation split, so a test data set of centroid languages is used in training, and the training set is used for evaluation (test set).

6.5.4 Evaluation Metrics

Both automatic and manual evaluation metrics are used to ensure the quality of the generated text. Particularly, for automatic evaluation **ROUGE-L** [Lin04b] and **BLEU**[4] [PRWZ02c] metrics are used for ATS and QG, respectively. Similar to Chi et al. [CDW$^+$20a], we used three manual evaluation metrics: **Fluency** referring to *how fluent the generated text is*, **Relatedness** indicating *the degree of the input's context in the generated text* and **Correctness** measuring the *grammar and semantics of generated text.* It is often observed that NLG systems suffer from the problem of Hallucination [NgYW$^+$19]; the *Relatedness* metric provides clarity in such situa-

[4] Reported scores are `case-mix BLEU-4` from modified sacreBLEU implementation. We modified the sacreBLEU and ROUGE-L to incorporate language-specific tokenizers and stammers for different languages.

Task/Dataset	Cluster-1		Cluster-2		Cluster-3		Centroid Lang	Non-Centroid Lang
	Lang	**MeanCD**	**Lang**	**MeanCD**	**Lang**	**MeanCD**	**Meta-train Lang**	**Target Lang**
Sum/XL-Sum	Punjabi	0.505	Spanish	0.253	Vietnamese	0.291	Punjabi	Tamil ,Marathi
	Tamil	0.547	Portuguese	0.437	Thai	0.326	Spanish	Gujarati , Bengali
	Marathi	0.548	French	0.477	Indonesian	0.327	Vietnamese	Telugu, Hindi
	Gujarati	0.550			Arabic	0.465		Nepali , Urdu
	Bengali	0.566			Chinese	0.561		Japanese, Turkish
	Telugu	0.574			Russian	0.902		Korean, Swahili
	Hindi	0.630						Portuguese, French
	Nepali	0.659						Thai, Indonesian
	Urdu	0.663						Arabic, Chinese
	Japanese	0.749						Russian
	Turkish	0.803						
	Korean	0.808						
	Swahili	-						
Sum/Wikilingua	Korean	0.558	Spanish	0.459	Vietnamese	0.484	Korean	Japanese, Turkish
	Japanese	0.583	French	0.476	Thai	0.496	Spanish	Hindi, French
	Turkish	0.620	German	0.529	Indonesian	0.536	Vietnamese	German, Portuguese
	Hindi	1.166	Portuguese	0.535	Arabic	0.595		Italian, Dutch
			Italian	0.566	Chinese	0.758		Thai, Indonesian
			Dutch	0.674	Russian	0.897		Arabic, Chinese
					Czech	1.374		Russian, Czech
QG/MLQA	Japanese	1.156	German	0.843	Vietnamese	0.299	Japanese	Hindi, Spanish
	Hindi	1.156	Spanish	0.843	Chinese	0.459	German	Chinese, Arabic
					Arabic	0.483	Vietnamese	
QG/TyDiQA	Telugu	0.682			Arabic	0.579	Telugu	Tamil, Bengali
	Tamil	0.719			Indonesian	0.619	Arabic	Finnish, Korean
	Bengali	0.769			Russian	0.940		Swahili, Indonesian
	Finnish	0.785						Russian
	Korean	0.828						
	Swahili	-						
QG/XQuAD	Turkish	1.038	Spanish	0.606	Thai	0.515	Turkish	Hindi, Romanian
	Hindi	1.038	Romanian	0.788	Arabic	0.516	Spanish	German, Arabic
			German	1.024	Vietnamese	0.519	Thai	Vietnamese, Chinese
					Chinese	0.813		Russian, Greek
					Russian	0.926		
					Greek	1.071		

Table 6.3: Details of language clustering for each dataset, mean cosine distance (meanCD), and centroid languages. For each dataset, we group languages into three clusters as shown in Figure 6.1. The Swahili language does not have any typological or task-based representations, so we added it to cluster 1 based on language typological features and heuristics. For the TyDiQA dataset, only two clusters are obtained as cluster-2 does not have any language. If a cluster has only two languages, we randomly selected any language as a centroid language.

tions. The *Correctness* metric is the hard metric that considers both semantic and grammatical aspects of generated text.

We randomly sampled 50 generated examples for each <task, dataset, language> triplet based on qualified and available native language experts in Hindi, Telugu, Tamil, and Bengali languages. In total, we selected six triplets for evaluation. To ensure the inter-annotator agreement and quality, each selected triplet is evaluated by two sets of annotators. We asked each annotator to rate the generated text on a scale of 1-5 (where one is very bad and five is very good) for the metrics mentioned above. We anonymously shared the generated text from two baselines and Meta-X$_{\text{XNLG}}$ to avoid any biased evaluation.

6.5.5 Implementation Details

We implemented Meta-X$_{\text{NLG}}$ using *higher* library[5]. SGD with learning rate (α) $1e-4$ is used as an inner-loop optimizer, and AdamW with learning rate (β) $1e-5$ is used as an outer-loop optimizer. The inner iteration (m) value is two, and the meta-training batch size is 8. To partition the training batch into support set (S) and query set (Q), we experimented (S: Q) with [80:20, 70:30, 60:40, 50:50, 40:60]% splits. The best results are obtained with equal partition, i.e., 50:50. We also experimented with [2, 5, 10, 15, 20, 25] training epochs. The best performance was observed at 10^{th} epoch. We use a standard mT5-small sequence-to-sequence Transformer architecture with 12 layers (each 16 heads). It has 1024 dimensions and approx 582M parameters. Additional layer-normalization with weight decay (0.1) was used with both the encoder and decoder. For input, the max sequence length is fixed to 512. We trained all the models on 1 Nvidia V100 GPU (32GB). Cross-entropy label smoothing is used as a loss function. We use beam-search with beam size 4; max generation length is 100 for ATS (32 for QG) and min length is 1. To ensure the stated improvement on the MLQA dataset, we compute average BLEU scores across the best five checkpoints. We are unable to repeat such experiments for other datasets due to computational limits.

6.6 Results and Analyses

Automated evaluation results are shown in Tables 6.4-6.8. Meta-X$_{\text{NLG}}$ consistently outperformed the other two baselines on all five datasets and most of the languages. For the summarization task, among the 33 experiments (19 languages for XL-Sum and 14 for Wikilingua) Meta-X$_{\text{NLG}}$ gives the best performance for 30 experiments. Wherever it loses out, it does so by a small margin. We see that the performance gains for the Wikilingua are relatively smaller. This might be due to the nature of the Wikilingua dataset; we observe that the input documents are a set of usage instructions for software/tools. For such data, many instructions need to be retained in the summary. This poses a challenge to all the models including Meta-X$_{\text{NLG}}$. Similar observations are made by [MDKD21a].

For the question generation task, Meta-X$_{\text{NLG}}$ exbits similar trends and outperforms both baselines across datasets and most of the LRLs except for one - the Indonesian language for TyDiQA. For MLQA, improvements achieved by the proposed model

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

are marginal (see Table-6.8). Upon close inspection, we notice that MLQA had a small number of languages, and the centroid languages are very distinct, i.e., they have a higher mean distance to other languages from the same cluster as compared to the other datasets (see Table-6.3). This might be a possible reason for such performance. The human evaluation scores for all three metrics are shown in Table-6.9. The human evaluations (across both annotator sets) correlate with automatic evaluations. Similar to the automatic evaluation, Meta-X$_{\text{NLG}}$ consistently outperformed both baselines for selected languages, tasks, and datasets. High *Fluency* and *Relatedness* scores for Meta-X$_{\text{NLG}}$ indicate that most of the generated text is fluent and not hallucinated respectively. The correctness metric considers both semantic and grammatical aspects; good scores on this metric indicate the acceptable performance for the proposed model in a zero-shot setting. In QG, generating well-formed interrogative sentences is challenging, particularly in zero-shot settings due to the unseen interrogative syntax structure of target language [MJVG21, MDKD21a]. The above-average fluency and correctness score for Meta-X$_{\text{NLG}}$ indicates that the model quickly adapts such syntax and performs better.

The consistent improvement in Meta-X$_{\text{NLG}}$ for most of the typologically diverse target languages provides evidence that supervision transfer is more uniform. Considering acceptable automatic and manual evaluation scores in the zero-shot setting, we conclude that our model performs reasonably well except small performance gain with the MLQA dataset. Meta-X$_{\text{NLG}}$ is a zero-shot framework, and we do not assume any prior training/knowledge for new unseen LRL. The only constraints are that the new language should be part of base pre-trained models (mT5) and adaptive unsupervised pre-training (uses task-agnostic monolingual data only). Hence, adding new languages in Meta-X$_{\text{NLG}}$ is a simple extension exercise.

Model	fr	gu	id	th	ta	hi	mr	ja	ko	tr	ru	sw	pt	ar	te	ur	ne	bn	zh
EnZmT5	18.45	13.21	19.77	21.53	11.58	22.24	11.89	22.81	18.74	17.72	15.27	18.91	18.92	18.44	10.77	21.61	**16.24**	16.12	21.07
FTZmT5	21.83	7.98	19.27	**24.68**	10.80	11.92	8.94	23.32	16.82	14.99	12.90	21.01	20.07	15.85	9.14	13.05	11.06	12.66	15.20
Meta-X$_{\text{NLG}}$	**22.83**	**14.02**	**21.54**	24.61	**12.88**	**23.09**	**12.58**	**25.33**	**20.12**	**18.65**	**17.31**	**22.63**	**20.24**	**20.11**	**12.07**	**23.41**	15.45	**17.96**	**22.95**

Table 6.4: Zero-shot ROUGE-L scores for 19 target LRL on XL-Sum dataset [HBI+21]. EnZmT5 [MDKD21a] and FTZmT5 are baseline models. Scores are reported after an extensive hyperparameter search for all the models.

6.6.1 Cross-lingual Transfer:

To have a more general view of the model's learning of multiple languages, we perform similarity analysis among representations of the language tags (contextual representa-

Model	**id**	**fr**	**ar**	**pt**	**it**	**th**	**ru**	**cs**	**nl**	**de**	**ja**	**zh**	**hi**	**tr**
EnZmT5	15.34	18.72	**15.70**	17.21	15.05	26.66	14.67	9.42	17.97	13.69	22.32	20.12	18.88	14.45
FTZmT5	13.69	19.37	12.66	17.80	15.54	23.72	11.95	10.20	16.74	12.22	22.81	18.64	17.32	13.84
Meta-X$_{\mathbf{NLG}}$	**16.85**	**20.26**	15.66	**18.36**	**16.03**	**27.71**	**14.89**	**11.76**	**19.09**	**14.11**	**22.83**	**22.45**	**19.60**	**15.23**

Table 6.5: Zero-shot ROUGE-L scores for 14 target LRLs on Wikilingua dataset [LDCM20a].

Model	**ar**	**de**	**zh**	**vi**	**hi**	**el**	**ru**	**ro**
EnZmT5	8.55	9.99	23.76	17.29	9.55	8.18	10.98	11.27
FTZmT5	5.82	9.040	22.87	16.47	9.05	6.95	8.87	10.31
Meta-X$_{\mathbf{NLG}}$	**8.63**	**10.52**	**24.89**	**20.92**	**11.90**	**9.01**	**11.41**	**12.24**

Table 6.6: Zero-shot BLEU scores for 8 target LRLs on XQuAD dataset [ARY20a].

tion of the `<fxx><2xx>` tokens from the beginning of the input in language `xx`). First, we randomly select 10 languages from the XL-Sum dataset. Then, each language input is passed through the trained encoder of the EnZmT5 baseline and Meta-X$_{\text{NLG}}$ to obtain language tag representations (LTRs). Finally, cosine distance among LTRs is computed and shown in figure-6.3. It can observed that the EnZmT5 baseline has a high cosine distance (dark colors) between LTRs and the shared latent representation space is not much aligned. Meta-X$_{\text{NLG}}$ has lower distances (light colors) and shared latent representation space is more aligned across languages.

6.6.2 Effect of Different Centroid (Meta-Train) Languages:

Table-6.10 shows the results with 36 different language combinations for Meta-X$_{\text{NLG}}$ training on the XL-Sum dataset. For this dataset, the centroid languages are Panjabi (pa), Spanish (es), and Vietnamese (vi). Results are generally good when centroid languages are in the training set. The best results are obtained using three centroid languages from three clusters. The performance dropped when we included fewer or more than three centroid languages. As discussed in section-6.4.1, learning gets

Model	**fi**	**ru**	**id**	**sw**	**ko**	**bn**	**ta**
EnZmT5	7.87	5.52	5.75	4.48	8.59	5.77	3.08
FTZmT5	8.39	7.28	**11.42**	5.51	10.05	7.96	2.022
Meta-X$_{\mathbf{NLG}}$	**9.08**	**7.47**	9.36	**6.42**	**12.67**	**9.17**	**9.76**

Table 6.7: Zero-shot BLEU scores for 7 target LRLs on TyDiQA data [CCC+20a].

Model	hi	es	ar	zh
EnZmT5	5.06	6.94	3.46	13.70
FTZmT5	5.14	6.16	2.21	13.38
Meta-X$_{\text{NLG}}$	**5.66**	**7.03**	**3.66**	**15.13**

Table 6.8: Zero-shot BLEU scores for 4 target LRLs on MLQA data [LOR+20a].

Model	Task/Data/Lang	Flu	Rel	Corr	Task/Data/Lang	Flu	Rel	Corr
	Annotator set-1							
EnZmT5	ATS/XL-Sum/bn	4.06	3.58	2.84	ATS/XL-Sum/te	4.28	3.94	3.70
FTZmT5		2.82	3.18	2.08		3.46	3.46	3.22
Meta-X$_{\text{NLG}}$		**4.12**	**4.34**	**3.44**		**4.50**	**4.22**	**4.04**
	Annotator set-2							
EnZmT5	ATS/XL-Sum/bn	3.70	3.23	3.26	ATS/XL-Sum/te	3.56	3.50	3.20
FTZmT5		2.62	2.48	2.16		3.02	2.84	2.60
Meta-X$_{\text{NLG}}$		**3.97**	**3.48**	**3.28**		**4.18**	**4.10**	**3.88**
	Annotator set-1							
EnZmT5	ATS/Wiki/hi	4.00	3.72	3.68	QG/XQuAD/hi	4.12	4.24	2.54
FTZmT5		4.07	3.39	3.83		4.22	4.02	2.56
Meta-X$_{\text{NLG}}$		**4.09**	**3.80**	**3.97**		**4.42**	**4.34**	**2.86**
	Annotator set-2							
EnZmT5	ATS/Wiki/hi	4.38	4.22	4.00	QG/XQuAD/hi	3.28	3.63	2.82
FTZmT5		4.57	**4.44**	4.08		3.24	3.34	2.89
Meta-X$_{\text{NLG}}$		**4.66**	**4.44**	**4.16**		**3.59**	**3.67**	**3.24**
	Annotator set-1							
EnZmT5	QG/MLQA/hi	3.48	3.70	3.46	QG/TyDiQA/ta	4.25	4.06	3.10
FTZmT5		3.44	3.42	3.18		3.25	3.01	2.07
Meta-X$_{\text{NLG}}$		**3.70**	**3.74**	**3.56**		**4.74**	**4.20**	**3.39**
	Annotator set-2							
EnZmT5	QG/MLQA/hi	**3.30**	3.28	2.40	QG/TyDiQA/ta	3.00	4.08	2.82
FTZmT5		3.10	3.44	2.84		2.55	3.045	1.83
Meta-X$_{\text{NLG}}$		3.24	**3.70**	**2.88**		**4.04**	**4.46**	**3.20**

Table 6.9: Human Evaluation results for four languages (**hi:** Hindi, **te:** Telugu, **ta:** Tamil and **bn:** Bengali), two annotator sets, two tasks (**ATS** and **QG**) and all five datasets. **Flu:** Fluency, **Rel:** Relatedness and **Corr:** Correctness metrics. Results are shown for two annotation sets, which ensure bias-free evaluation. Reported scores are the average of all the annotators in an annotator set.

distracted with many centroid languages. Overall, Meta-X$_{\text{NLG}}$ trained with three centroid languages (row 36) performs best on most of the languages and on average.

6.6.3 Case Study

Fig. 6.4 presents zero-shot generations from Meta-X$_{\text{NLG}}$ in Telugu, Tamil, Bengali and Hindi languages for both ATS and QG tasks. With this qualitative analysis, we can observe that the zero-shot generation is high quality and acceptable.

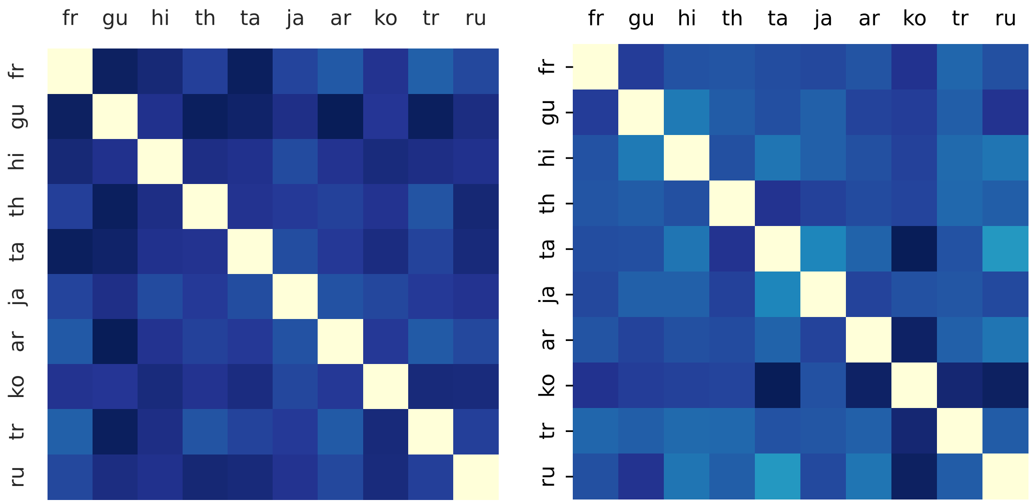


Figure 6.3: Cosine distance between language tags obtained from EnZmT5 baseline (left) and Meta-X$_{\text{NLG}}$ (right) for 10 languges from XL-Sum dataset. Dark colors indicate a higher cosine distance.

SetUp	Meta-Train Langs	fr	gu	id	th	ta	hi	mr	ja	ko	tr	ru	sw	pt	ar	te	ur	ne	bn	zh	avg
1*	pa	16.59	7.55	15.87	23.57	11.10	13.22	9.54	24.17	17.67	15.61	13.51	17.34	16.42	15.94	9.19	12.69	11.84	13.25	20.71	15.04
2*	es	21.35	12.73	19.54	23.82	10.42	18.77	10.99	24.15	18.02	15.87	14.10	20.03	19.72	17.46	10.13	20.12	15.06	16.00	22.01	17.38
3*	vi	19.67	12.34	18.69	25.02	11.05	19.41	10.90	23.77	18.46	15.15	14.56	20.40	18.02	17.43	10.69	20.23	14.42	15.47	21.58	17.22
4*	ru	17.60	12.89	16.97	23.54	10.50	18.03	10.75	24.28	18.09	16.36	-	18.25	17.32	17.63	10.44	20.52	14.28	14.40	22.18	16.89
5*	tr	16.57	12.83	16.04	23.77	10.10	17.72	10.65	24.06	17.01	-	14.90	19.46	17.34	17.59	10.40	20.12	13.51	13.35	21.01	16.46
6**	np	16.89	9.23	16.47	23.44	10.70	21.51	10.45	24.73	17.12	15.28	14.16	17.03	16.54	16.03	10.43	19.21	-	13.28	21.81	16.35
7**	th	17.86	11.60	17.25	-	10.78	17.98	10.30	21.07	17.89	15.73	14.48	18.16	17.59	17.19	9.87	20.11	13.56	15.65	15.35	15.69
8*	vi, pa	19.50	7.98	18.02	24.41	11.25	13.33	9.45	23.96	17.37	15.09	13.61	19.34	17.99	16.13	9.11	14.05	11.93	13.20	18.91	15.51
8*	tr, es	21.40	12.55	19.73	23.75	11.65	20.61	10.71	24.92	19.28	-	14.12	20.11	19.44	17.17	11.74	21.40	14.78	16.54	22.82	17.93
10*	fr, vi	-	12.49	19.51	23.72	11.12	18.83	10.38	24.01	18.74	15.98	14.01	19.40	18.96	17.18	10.52	20.44	14.32	15.19	22.36	17.06
11**	ur, zh	18.06	12.56	17.26	22.30	11.95	14.27	11.53	21.40	18.51	17.02	14.73	17.58	17.20	17.76	11.18	-	14.41	15.98	-	16.10
12**	th, pt	21.28	12.39	19.60	-	10.83	17.90	10.04	22.49	17.02	16.07	14.52	20.19	-	17.61	10.00	19.79	13.77	15.10	21.45	16.47
13@	pa, pt	21.13	8.72	19.92	23.89	11.64	14.38	9.65	24.13	17.36	16.89	14.91	20.90	-	17.36	9.95	15.53	11.66	13.37	22.04	16.30
14@	es, bn	21.61	10.53	18.85	23.23	11.06	17.33	10.15	24.31	17.25	15.68	13.69	19.32	19.27	16.29	10.46	20.40	11.75	-	19.48	16.70
15*	pa,fr,ru	-	9.80	19.17	23.39	10.54	13.97	9.43	24.41	17.50	16.56	-	19.52	19.07	16.08	9.03	16.44	11.43	13.01	21.71	15.95
16*	pa,es,ru	21.34	9.42	19.04	24.58	10.67	13.17	9.02	24.04	16.92	16.30	-	19.90	19.60	16.20	8.98	14.97	11.86	12.76	21.89	16.15
17*	vi,pa,fr	-	9.75	19.31	23.65	11.18	13.98	9.41	24.52	17.91	15.88	13.79	20.20	19.24	16.28	9.47	15.68	11.78	13.75	19.48	15.85
18**	ko,pt,th	21.66	12.94	19.93	-	11.94	20.35	10.42	24.46	-	17.99	15.55	21.22	-	18.58	11.23	21.54	15.20	16.06	16.72	17.24
19**	gu,pt,ar	21.83	-	19.52	23.74	10.30	14.46	7.71	23.51	15.57	15.34	13.73	19.40	-	-	9.62	18.77	11.30	12.88	21.03	16.17
20@	es,th,ar	22.11	12.14	19.60	-	10.60	17.22	9.92	22.88	16.78	16.18	13.81	20.42	20.09	-	10.25	19.55	13.58	15.35	17.27	16.34
21@	pa,pt,vi	21.75	9.65	19.80	24.49	11.41	13.82	9.81	24.51	17.70	16.16	14.55	20.39	-	17.28	10.04	15.71	11.70	13.91	20.97	16.31
22*	pa,es,vi,fr	-	9.35	19.74	23.91	11.11	13.86	8.96	24.82	17.70	16.54	13.57	20.65	20.16	16.43	9.52	16.76	11.73	13.48	19.81	16.01
23*	pa,ep,vi,ru	21.90	8.39	19.28	**24.89**	10.65	14.19	9.38	24.25	16.47	16.00	-	21.20	20.12	16.38	9.19	16.07	11.62	12.98	19.03	16.06
24*	pa,es,vi, tr	22.35	9.89	20.57	24.59	11.45	15.10	9.59	25.44	17.70	-	13.89	21.55	20.28	17.23	10.00	17.20	12.73	13.58	19.82	16.83
25**	zh,bn,te,pt	21.73	10.94	18.98	22.99	10.58	16.23	9.46	20.57	16.16	15.80	13.57	20.23	-	16.23	-	19.51	12.23	-	-	16.35
26**	id,sw,ur,pt	22.70	12.77	-	24.17	10.95	15.94	10.68	24.77	17.58	17.13	14.42	-	-	18.64	10.39	-	13.70	14.40	22.87	16.74
27@	pa,es,vi,hi	21.81	8.66	19.21	24.43	10.64	-	11.03	24.25	17.20	16.12	12.89	20.86	19.93	16.25	9.58	16.15	**16.36**	12.56	13.78	16.21
28@	pa,es,vi,ko	22.33	12.47	20.70	23.70	12.53	19.55	10.75	25.44	-	17.90	15.02	**22.63**	19.97	18.33	11.68	21.52	14.71	16.26	21.32	18.16
29*	pa,es,vi,fr,tr	-	10.26	20.39	24.04	11.12	14.79	9.08	25.42	17.75	-	13.35	21.17	20.28	16.50	9.65	17.43	12.43	14.01	20.62	16.37
30*	pa,es,vi,ru,mr	21.77	10.12	19.44	23.85	10.81	23.85	-	24.20	16.95	16.02	-	20.60	19.97	16.30	9.57	17.46	15.71	13.47	18.40	17.56
31**	id,sw,ur,po,te	22.43	11.19	-	23.88	9.87	16.08	9.64	24.21	16.05	17.05	14.19	-	-	18.54	-	-	13.08	13.19	20.44	16.42
32@	pa,es,te,mr,gu	20.51	-	18.05	22.01	9.69	**23.94**	-	21.93	15.32	15.04	11.83	18.51	19.39	14.60	-	16.70	15.81	12.70	10.13	16.63
33*	pa,es,vi,fr,tr,ru	-	9.98	20.59	24.61	11.14	14.72	9.21	25.18	17.53	-	-	21.54	20.55	16.61	9.65	17.72	12.07	13.71	21.80	16.66
34*	pa,es,vi,fr,tr,ru,mr	-	10.15	20.65	24.42	10.56	24.34	-	24.66	17.09	-	-	21.28	20.60	16.11	9.97	18.21	15.81	13.21	19.32	17.76
35*	pa,es,vi,fr,tr,ru,mr,ja	-	9.88	19.61	23.51	9.83	23.40	-	-	13.27	-	-	21.43	**20.36**	15.83	9.24	15.66	16.24	12.68	20.32	16.52
36*	**Meta-X$_{\text{NLG}}$(pa,es,vi)**	**22.83**	**14.02**	**21.54**	24.61	**12.88**	23.09	**12.58**	25.33	**20.12**	**18.65**	**17.31**	**22.63**	20.24	**20.11**	**12.07**	**23.41**	15.45	**17.96**	**22.95**	**19.40**

Table 6.10: Meta-X$_{\text{NLG}}$'s zero-shot evaluation scores (Rouge-L) with different meta-training (centroid) language combinations on the XL-Sum dataset. We cut the hierarchical clustering dendrogram shown in Figure 6.2, at the lower level to obtain more clusters. In total, we obtained eight centroid languages, i.e., pa, es, vi, tr, ja, mr, fr and ru. '-' indicates the language used in training, so scores are not zero-shot and not included. Markers '*', '**', and '@' indicate meta training with all-centroid, all-non-centroid, and a mix of both (centroid & non-centroid) languages.

6.7 Conclusion

In this work, we propose a novel Meta-X_{NLG} framework based on meta-learning and language clustering for effective and more uniform cross-lingual transfer and generation. To the best of our knowledge - this is the first study that uses meta-learning for zero-shot cross-lingual transfer and generation. The evaluations are done with two downstream challenging NLG tasks (ATS and QG), five publicly available datasets and 30 languages. Consistent improvement for both human and automatic evaluation metrics is observed over baselines. The cross-lingual transfer analysis indicates the model's ability towards uniform cross-lingual transfer to multiple low-resource languages.

6.8 Insights, Limitations and Future Work

Insights: As discussed, there is a trade-off between the number of clusters and generalization capabilities. To ensure that we have selected the correct number of clusters, we conducted an extensive ablation study with 36 experimental setups involving different numbers of clusters and various combinations of languages. We observed that the model with three clusters performs the best. The language cluster obtained with the approach proposed by Oncevay et al. [OHB20] resulted in clustering that is close to the clustering approach with language family - further validating the correctness of clustering. Unlike the ZmBART zero-shot QG model, where generated questions are of a code-mixed nature, starting with *wh-words*, the Meta-X_{NLG} model successfully mitigates these issues and generates well-formed questions. This indicates the adaptability of Meta-X_{NLG} for different language structures through meta-learning.

Limitations:

The Meta-X_{NLG} framework has two limitations: (1) Similar to the ZmBART model for new languages, there is a need to re-train the adaptive unsupervised step with new languages. (2) We require small, task-specific annotated (validation) data for centroid languages, which will be used in the meta-training.

Future Work: In the future, we will extend this framework to more languages, tasks, and datasets. We will also plan to advance language technology for those LRLs that do not have parallel data, possess limited monolingual data, and whose representations are absent from large multilingual pre-trained language models (Chapter 7).

Telugu-XLSum

Input Document: ప్రభుత్వంలో ఆర్టీసీ విలీనం సహా తమ డిమాండ్లన్నింటినీ సాధించే వరకూ పోరాటం ఆపబోమని కార్మికులు ప్రకటించారు. ఆర్టీసీ కార్మిక సంఘాల జేఏసీ ఆధ్వర్యంలో ఈ సభ జరిగింది. కార్మికులకు మద్దతుగా పలు రాజకీయ పార్టీల నాయకులు దీనికి హాజరయ్యారు. సభ జరిగిన సరూర్‌నగర్ ఇండోర్ స్టేడియం కార్మికులతో నిండిపోయింది. కాంగ్రెస్ నాయకుడు రేవంత్ రెడ్డి, టీజేఎస్ అధ్యక్షుడు కోదండ రాం, టీడీపీ తెలంగాణ అధ్యక్షుడు ఎల్.రమణ, సీపీఐ నాయకుడు చాడ వెంకట్ రెడ్డి, బీజేపీ నేత వివేక్, ఎంఆర్పీఎస్ నాయకుడు మంద కృష్ణమాదిగతోపాటు పలు ప్రజా సంఘాలు, రాజకీయ పార్టీల నాయకులు, కళాకారులు, కార్మిక సంఘాల ప్రతినిధులు ఈ సభకు హాజరయ్యారు. సభలో మాట్లాడిన వారంతా ప్రభుత్వ వైఖరిని తప్పు పట్టారు. కార్మికులకు అండగా ఉంటామని భరోసా ఇచ్చారు. ఆర్టీసీని విలీనం చేయడం ఎందుకు సాధ్యం కాదో చెప్పాలని రేవంత్ రెడ్డి ప్రభుత్వాన్ని డిమాండ్ చేశారు. తెలంగాణ సీఎం కేసీఆర్ తీసుకునే నిర్ణయాలు అన్నీ మేనిఫెస్టోలో పెట్టే తీసుకుంటున్నారా అని ప్రశ్నించారు. కార్మికులకు మద్దతుగా ఆందోళనలు చేస్తామని, అవసరమైతే మిలియన్ మార్చ్ నిర్వహిస్తామని కోదండరామ్ హెచ్చరించారు. "కేసీఆర్ ఎన్నో పుస్తకాలు చదివానని చెప్పుకుంటాడు. కానీ, ఆయనకు రాజ్యాంగంపై కూడా అవగాహన లేదు" అని చాడ వెంకటరెడ్డి విమర్శించారు. ఈ సమ్మె విషయంలో కేంద్రం జోక్యం చేసుకోవాలని ఆయన కోరారు. కార్మికుల ఐక్యతను దెబ్బతీయడానికి కేసీఆర్ కుట్రలు పన్నుతున్నారని, కార్మికులు అప్రమత్తంగా ఉండాలని ఎల్.రమణ విజ్ఞప్తి చేశారు. ఆర్టీసీని అన్యాక్రాంతం చేయాలని ప్రభుత్వం ప్రయత్నాలు చేస్తోందని ఆరోపించారు. కేసీఆర్ అవకాశవాది అని, తప్పుడు ప్రచారాలు చేస్తున్నారని బీజేపీ నేత వివేక్ అన్నారు. సకల జన భేరి సభకు హైదరాబాద్‌తో పాటు ఇతర ప్రాంతాలకు చెందిన ఆర్టీసీ సిబ్బంది తరలి వచ్చారు. సభలో కేసీఆర్‌కు వ్యతిరేకంగా నినాదాలు చాలా సార్లు వినిపించాయి. సభా ప్రాంగణం సరిపోక పోవడంతో బయట కూడా పెద్ద సంఖ్యలో కార్మికులు నిలబడిపోయారు. ఆర్టీసీ కార్మికులతో పాటు వామపక్షాల కార్యకర్తలు కూడా సభకు పెద్ద సంఖ్యలో హాజరయ్యారు. ఆర్టీసీ సమ్మె బుధవారానికి 26 రోజులకు చేరుకుంది. ప్రస్తుతానికి సమ్మె ముగిసే సంకేతాలు ఏమీ కనిపించడం లేదు. ప్రభుత్వం, కార్మిక సంఘాలు తమ తమ వాదనలకు కట్టుబడి ఉన్నాయి. ఆర్టీసీపై ముఖ్యమంత్రి కేసీఆర్ బుధవారం సమీక్ష నిర్వహించారు. మరోవైపు, గురువారం అన్ని డిపోల ముందు నిరాహార దీక్షలు చేయాలని ఆర్టీసీ జేఏసీ పిలుపునిచ్చింది. ఇవి కూడా చదవండి: (బీబీసీ తెలుగును ఫేస్‌బుక్, ఇన్‌స్టాగ్రామ్, ట్విటర్‌లో ఫాలో అవ్వండి. యూట్యూబ్‌లో సబ్‌స్క్రైబ్ చేయండి.)

Human: సమ్మె బాట పట్టిన తెలంగాణ ఆర్టీసీ కార్మికులు బుధవారం హైదరాబాద్ లో సకల జనభేరి పేరుతో సభ నిర్వహించారు.

Meta-X$_{NLG}$: తెలంగాణ ఆర్టీసీ కార్మిక సంఘాల జేఏసీ ఆధ్వర్యంలో జరిగిన సకల జన భేరి సభ హైదరాబాద్ లో జరిగింది.

Bengali-XLSum

Input Document: ভারতের অন্য অঞ্চলেও কোক, পেপসি নিষিদ্ধ করার দাবি জানাচ্ছেন কর্মীরা। স্থানীয় পণ্যের ব্যবহার নিশ্চিত করার জন্যই এই উদ্যোগ গ্রহণ করেছে ব্যবসায়ীরা। রাজ্যের শীর্ষ দুটি ব্যবসায়ী এসোসিয়েশন এই দুটি পানীয় নিষিদ্ধ করার প্রস্তাব করেছিল। তারই প্রেক্ষাপটে আজ বুধবার থেকে তামিলনাড়ু রাজ্যে নিষিদ্ধ হলো কোকা-কোলা ও পেপসি। প্রতিষ্ঠানগুলো বলছে, কোমল পানীয়ের প্রতিষ্ঠানগুলো নদী থেকে প্রচুর পানি ব্যবহার করে, সেকারণে কৃষকদের জমি সেচের সময়ও ব্যাপক ভোগান্তিতে পড়তে হয়। বিশেষ করে খরার সময় সেচে পানি সমস্যা প্রকট হয়ে দাঁড়ায়। রাজ্যের দশ লাখেরও বেশি দোকানদার এ নিষেধাজ্ঞা মেনে চলবে বলে ধারণা করা হচ্ছে। গত মাসে তামিলনাড়ুতে 'জাল্লিকাটু' নামে ঐতিহ্যবাহী ষাঁড়ের লড়াই নিষিদ্ধের বিরুদ্ধে ব্যাপক বিক্ষোভের ঘটনা দেখে রাজ্যে পেপসি, কোকা-কোলা নিষিদ্ধের প্রস্তাব করে শীর্ষ দুটি ব্যবসায়ী সংগঠন ফেডারেশন অব তামিলনাড়ু ট্রেডার্স এসোসিয়েশন (এফটিএনটিএ) এবং তামিলনাড়ু ট্রেডার্স এসোসিয়েশন। বিক্ষোভের সময় অনেকে বলছিলেন 'জাল্লিকাটু' নিষিদ্ধ করা মানে স্থানীয় ঐতিহ্য ও সংস্কৃতিকে অবমাননা করা। \"আমরা কয়েক মাস আগে কোমল পানীয়ের বিরুদ্ধে আমাদের প্রচারণা শুরু করি, কিন্তু যখন আমরা 'জাল্লিকাটু' নিষিদ্ধের প্রতিবাদে বিক্ষোভ শুরু করি, কোমল পানীয়ের বিরুদ্ধে আমাদের প্রচারণাও ভিন্ন রূপ পায়\"- বিবিসি তামিল সার্ভিসকে দেয়া এক সাক্ষাৎকারে বলছিলেন এফটিএনটিএ'র প্রেসিডেন্ট থা ভেলায়ান। \"পেপসি"

Human: ভারতের দক্ষিণাঞ্চলীয় রাজ্য তামিলনাড়ুর ব্যবসায়ীরা সেখানে কোকা-কোলা ও পেপসি বিক্রি নিষিদ্ধ ঘোষণা করেছে।

Meta-X$_{NLG}$: ভারতের তামিলনাড়ু রাজ্যে কোমল পানীয় নিষিদ্ধ করার দাবি জানিয়েছে ব্যবসায়ীরা।

Hindi-Wikilingua

Input Document: ठंडे पानी से धोते हुए दाग को कुछ हल्का करने की कोशिश करें। ठंडे पानी और एक टॉवल की मदद से धब्बे को गीला कर लें। या फिर, दाग को ठंडे पानी के नीचे भी लगाया जा सकता है। पेस्ट बनाने के लिए एक भाग ठंडा पानी और दो भाग नमक को मिलाएँ। आपको कितने नमक और पानी की जरूरत पड़ने वाली है, ये तो पूरी तरह से उस धब्बे के आकार पर निर्भर करेगा। नमक के साथ में बहुत ज्यादा पानी भी न मिलाएँ, नहीं तो ये लिक्विड बन जाएगा। पेस्ट फैलाने लायक गाढ़ा होना चाहिए। आप चाहें तो पेस्ट को दाग पर लगाने के लिए अपने हाँथ का या एक साफ कपड़े का यूज भी कर सकते हैं। पेस्ट को बहुत आराम से धब्बे के ऊपर रगड़ लें। अब आपको वो दाग हल्का होता हुआ नजर आना चाहिए। जैसे ही, ज़्यादातर या पूरा धब्बा निकल आए, कपड़े को ठंडे पानी के नीचे लगा दें। पेस्ट के पूरे साफ होने तक इसे धोते रहें। अगर दाग अभी तक पूरा नहीं निकल पाया है, तो पेस्ट को फिर से लगा दें। उस कपड़े को धोने के लिए भी उसी साबुन का यूज करें, जिसे आप नॉर्मली यूज किया करते हैं। हालाँकि, कपड़े को धोने के लिए ठंडे पानी के अलावा और किसी चीज़ का यूज बिल्कुल न करें। कपड़े को धो लेने के बाद, इसे हवा में सूखने के लिए लटका दें।

Human: दाग को ठंडे पानी से धो लें: नमक और पानी से एक पेस्ट तैयार करना: पेस्ट को धब्बे पर लगा लें: कपड़े को ठंडे पानी में धो लें: नॉर्मल जैसे ही धो लें:.

Meta-X$_{NLG}$: धब्बे को ठंडे पानी से धोने के लिए एक टॉवल का यूज कर लें। नमक और पानी का घोल बनाये। धब्बे को घोल से धोये; ठंडे पानी से धोये।

TyDiQA-Tamil

Passage: Coordinates: வால்ட் டிஸ்னி உலகம் (Walt Disney World) அல்லது வோல்ட் டிஸ்னி உலக ஓய்விடம் சுருக்கமாக டிஸ்னி உலகம் என்பது உலகின் மிக அதிகமானோர் செல்லும் சுற்றுலா மற்றும் பொழுதுபோக்கு ஓய்விடமாகும். இது அமெரிக்காவின் புளோரிடாவில் உள்ள பியூனா விஸ்டா என்ற ஏரியில் அமைந்துள்ளது.[1] வால்ட் டிஸ்னி நிறுவனத்தால் பராமரிக்கப்படும் இவ்விடத்தின் பரப்பளவு 30,080 ஏக்கர் (12,173 ஹெக்டயர்; 47 சதுர மைல்) பரப்பளவைக் கொண்டது. வால்ட் டிஸ்னி உலகில் நான்கு கேளிக்கைப் பூங்காக்களும் மற்றும் இரண்டு நீர்ப் பூங்காக்களும், இருப்பத்து நான்கு ஓய்வு விடுதிகளும் மற்றும் இரு ஆரோக்கிய நீரூற்று மற்றம் உடற்பயிற்சி நிலையங்கள், ஐந்து கோல்ப் விளையாட்டிடங்கள் மற்றும் பிற பொழுதுபோக்கு அம்சங்களும் உள்ளன.மேற்கோள்கள் வெளியிணைப்புக்கள்பகுப்பு:சுற்றுலாபகுப்பு:புளோரிடா

Answer: புளோரிடாவில்

Question (Human): டிஸ்னி வேர்ல்ட் எங்கு உள்ளது?

Question (Meta-X$_{NLG}$): வால்ட் டிஸ்னி உலகம் எங்கே அமைந்துள்ளது?

XQuAD-Hindi

Passage: दक्षिणी कैलिफोर्निया एक संयुक्त सांख्यिकीय क्षेत्र, आठ महानगरीय सांख्यिकीय क्षेत्रों, एक अंतरराष्ट्रीय महानगरीय क्षेत्र और कई महानगरीय डिवीजनों से मिलकर बना हुआ है। इस क्षेत्र में दो विस्तारित महानगरीय क्षेत्र बसे हुए हैं जो जनसंख्या में पांच मिलियन से अधिक हैं। इनके अंतर्गत ग्रेटर लॉस एंजिल्स क्षेत्र में 17,786,419, और सैन डिएगो-तिजुआना में 5,105,768 की आबादी हैं। इन महानगरीय क्षेत्रों में से, लॉस एंजिल्स-लॉन्ग बीच-सांता एना महानगरीय क्षेत्र, नदी के किनारे पर स्थिति-सैन बर्नार्डिनो-ओंटारियो महानगरीय क्षेत्र, और ऑक्सनार्ड-थाउज़ेंड ओक्स-वेंचुरा महानगरीय क्षेत्र मिलकर ग्रेटर लॉस एंजिल्स की रचना करते हैं; जबकि एल सेंट्रो महानगरीय क्षेत्र और सैन डिएगो-कार्ल्सबैड-सैन मार्कोस महानगरीय क्षेत्र दक्षिणी सीमा क्षेत्र बनाते हैं। ग्रेटर लॉस एंजिल्स के उत्तर में सांता बारबारा, सैन लुइस ओबिसपो और बेकर्सफील्ड महानगरीय क्षेत्र आते हैं।

Answer: 17,786,419

Question (Human): ग्रेटर लॉस एंजिल्स क्षेत्र की जनसंख्या कितनी है?

Question (Meta-X$_{NLG}$): ग्रेटर लॉस एंजिल्स क्षेत्र में कितनी आबादी है?

MLQA-Hindi

Passage: शिकागो विश्वविद्यालय के परिसर की पहली इमारतें, जो अब मुख्य प्रांगण के रूप में जानी जाती हैं, एक \"मास्टर प्लान\" का हिस्सा थीं, जिसकी कल्पना शिकागो विश्वविद्यालय के दो ट्रस्टियों द्वारा की गई थी और जिसे शिकागो के वास्तुकार हेनरी इवेस कॉब द्वारा तैयार किया गया था। मुख्य प्रांगण में छह चौकोर प्रांगण हैं, प्रत्येक प्रांगण एक चौकोर भवन से घिरा होता है, जिसके द्वारा एक बड़े चौकोर प्रांगण की सीमा बनती है। मुख्य प्रांगण की इमारतों को कोब, शेप्ली, रुटान और कूलिज, होलाबर्ड और रोश और अन्य वास्तुकला फर्मों द्वारा डिजाइन किया गया था, जो विक्टोरियन गोथिक और कॉलेजिएट गोथिक शैलियों के मिश्रण के रूप में ऑक्सफोर्ड विश्वविद्यालय के कॉलेजों पर आधारित हैं। (उदाहरण के लिए, मिशेल टॉवर, ऑक्सफोर्ड के मैग्डलेन टॉवर के बाद तैयार की गई है, और यूनिवर्सिटी कॉमन्स, हचिंसन हॉल, क्राइस्ट चर्च हॉल की प्रतिलिपि हैं।

Answer: मुख्य प्रांगण

Question (Human): विश्वविद्यालय द्वारा निर्मित पहली इमारत आज किस नाम से जानी जाती हैं?"

Question (Meta-X$_{NLG}$): शिकागो विश्वविद्यालय के परिसर की पहली इमारत का क्या नाम है?

Figure 6.4: Zero-shot samples generated by Meta-X$_{NLG}$ in Telugu, Tamil, Bengali and Hindi languages. The top three samples are for ATS, and the bottom three are for QG tasks. The generated samples are taken from all five datasets. In some instances, the model learns to generate an actual target language script even though the reference is in transliterated form. See the underlined token (in red font color) in the TyDiQA-Tamil example.

Chapter 7

Zero-Shot Machine Translation for Extremely Low-resource Languages

7.1 Introduction

In this chapter, we continue exploring cross-lingual modeling for low-resource languages (LRLs). The efforts with ZmBART, Meta-$\mathrm{X_{NLG}}$, and the NLP research community on multilingual modeling have extended the coverage of NLP technologies for many LRLs. However, there is a *long-tail* of languages for which there is no parallel/pseudo-parallel data, no/limited monolingual data, and their representations from the multilingual pre-trained language models (mPLMs) are absent. These languages are referred as *extremely low-resource languages (ELRLs).* Now, we turn our focus to enabling technology for ELRLs. For any language, the technology advances with a high-quality evaluation set that assesses the model's performance or tracks its progress. Most of the NLG tasks lack such a gold standard evaluation set for ELRLs. Considering this, in this chapter, we shifted our focus to the machine translation task where the evaluation test for ELRLs is available from the recently released FLORES-200 evaluation set [CjCÇ+22]. Before we dive into modeling details, let's take a step back and understand the state of language technology for ELRLs.

Recently, there has been remarkable progress in NLP research, primarily due to advancements in large pre-trained language models. The global linguistic landscape comprises approximately 7,000 spoken languages worldwide[1]. A notable disparity is evident in NLP research, with the majority of studies conducted on English data [Ben19, JSB+20]. This is concerning as the vast majority of the global population —

[1] https://www.ethnologue.com/insights/how-many-languages/

roughly 95% — does not speak English as their primary language, and a staggering 75% do not speak English at all[2]. According to Ruder et al. [Rud22], out of the 7,000 languages, approximately 400 languages have more than 1 million speakers and about 1,200 languages have more than 100,000 speakers. Despite this, only around 100 languages are incorporated into large pre-trained models, and limited resources are available for building NLP models for LRLs. Furthermore, a study presented at ACL 2008 [Ben11] revealed that 63% of all papers focused only on the English language. A more recent study during ACL 2021 [RVS22] concluded that nearly 70% of the papers were evaluated in English. Even a decade later, there has been little change and less focus on ELRLs.

The modern neural machine translation [AJF19, GSFP21, SBF^{+}22] has achieved remarkable performance for many languages, but their performance heavily relies on the availability of large parallel or monolingual corpora [KK17]. However, as mentioned earlier, ELRLs present unique challenges for model development. Evan for the MT task, the ELRLs lack parallel data, are excluded from mPLM, and possess limited monolingual data. Towards these concerns, **this work is positioned as a step towards enabling machine translation from ELRLs to English direction with no and limited resources**. Primarily focused on *zero-shot* setting for scalability.

Fortunately, many ELRLs are lexically similar to some HRLs. *Lexical similarity refers to languages sharing words with similar form (spelling and pronunciation) and meaning.*[3] This includes cognates, lateral borrowings and loan words. We explore if cross-lingual transfer can be enabled or improved for ELRLs by *explicitly* taking lexical similarity into account. In particular, **we explore MT from an ELRL to another language (English) with transfer enabled by a related HRL on the source side.** Our key *insight* is that cognates in ELRL having similar spelling to the HRL word can be thought of as misspellings of the latter. For example, the word "Monday" is *somvar* in the Hindi language and *somar* in the Bhojpuri language. They are lexically very similar. If we make the HRL to English MT model robust to spelling variations, it will improve cross-lingual transfer to related ELRLs. To achieve spelling variation robustness, we propose two novel *noise augmentation* approaches in the HRL of the HRL to English large parallel training data. The noise acts as a regularizer, and a model trained with this noisy HRL to English parallel data

[2] https://www.ethnologue.com/insights/most-spoken-language/

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

exhibits robustness to perturbations in representations of words in closely related ELRLs, hence improving model generalization.

Next, we will briefly provide an overview of the two proposed novel noise augmentation models: (1) Character-span noise augmentation and (2) Selective unsupervised noise augmentation.

Character-Span Noise Augmentation: In this modeling approach, we introduce random noise to each HRL example by adding 1 to 3 character grams (spans). Specifically, between 9% and 11% of the total characters in each example are selected noise augmentation. For noise augmentation, we employ *span deletion* and *span replacement with a single random character of ELRL* operations, both with equal probability. As the noise augmentation is based on character span, it is called as CharSpan model. In CharSpan, we do not require any data in ELRLs; only alphabets are used. Fig. 7.1 illustrates a sample example where the surface-level text alignment improves with span noise augmentation.

HRL (HIN):	इस सीज़न में बीमारी के शुरुआती मामले जुलाई के आखिर में सामने आए थे।
ENG:	The initial cases of the disease this season were reported in late July.
HRL (HIN)+CSN:	**ए_** सीज़न **म** बीमारी के **__प_** मामले जुलाई के आखिर **म** सामने आए **_**।
ELRL1 (BHO):	ए सीजन में ई बीमारी क पहिला मामला जुलाई क आखिर में सामने आ गइल रहले।
ELRL2 (HNE):	ए सीजन म ए बीमारी के पहिला मामला जुलाई के आखिर म सामने आए रहिस।

Figure 7.1: Hindi (HIN; HRL), Bhojpuri (BHO; ELRL) and Chhattisgarhi (HNE; ELRL) parallel sentences. Additionally, the corresponding noisy Hindi example with character-span (CSN) noise. BHO and HNE are closely related to HIN. After noise augmentation, noisy Hindi becomes lexically more similar to BHO and HNE.

Selective Unsupervised Noise Augmentation: Unlike CharSpan, this model is based on single character noise augmentation and noise augmentation is systematic (linguistically inspired) not random. It consists of two stages: *Selective Candidate Extraction* and *Noise Augmentation*. In the selective candidate extraction phase, candidate characters are extracted in an unsupervised manner using small monolingual data from closely related HRL and ELRLs. It relies on *BPE merge operations* and *edit operations* that take into account lexical similarity and linguistic properties. In the noise augmentation phase, noise is augmented into the source side of parallel data of HRL using greedy, top-k, and top-p sampling algorithms. The proposed model is referred to as the SelectNoise. SelectNoise, required small monolingual (1000 examples) data in ELRLs. Fig. 7.2 illustrates a sample example where the surface-level text alignment improves with proposed SELECTNOISE noise augmentation as compared to random character noise augmentation.

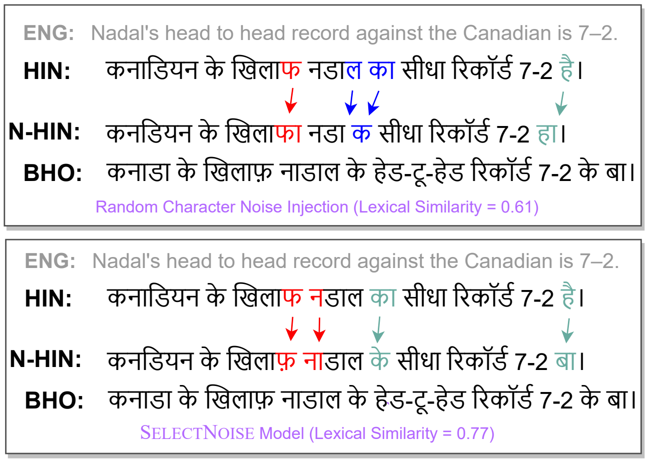

ENG: Nadal's head to head record against the Canadian is 7–2.
HIN: कनाडियन के खिलाफ नडाल का सीधा रिकॉर्ड 7-2 है।
N-HIN: कनडियन के खिलाफा नडा क सीधा रिकॉर्ड 7-2 हा।
BHO: कनाडा के खिलाफ़ नाडाल के हेड-टू-हेड रिकॉर्ड 7-2 के बा।
Random Character Noise Injection (Lexical Similarity = 0.61)
ENG: Nadal's head to head record against the Canadian is 7–2.
HIN: कनाडियन के खिलाफ नडाल का सीधा रिकॉर्ड 7-2 है।
N-HIN: कनडियन के खिलाफ़ नाडाल के सीधा रिकॉर्ड 7-2 बा।
BHO: कनाडा के खिलाफ़ नाडाल के हेड-टू-हेड रिकॉर्ड 7-2 के बा।
SELECTNOISE Model (Lexical Similarity = 0.77)

Figure 7.2: Illustration of character noise augmentation with random baseline [AS22] and propose SELECTNOISE model. The SELECTNOISE enhances lexical similarity between noisy HRL (N-HIN) and ELRL (BHO). ENG: *English*, HIN: *Hindi*, N-HIN: *Noisy Hindi* and BHO: *Bhojpuri* languages. Red: *Insertion*, Blue: *Deletion* and Green: *Substitution* operations.

CHARSPAN and SELECTNOISE models are explored dis-jointly at the parallel span of time. Consequently, the experimental setups and evaluations for these models differ from each other, with minimal overlap. Since both models address the same problem, we have presented them in a single chapter to avoid redundancy. However, it's important to note that we have not included any direct comparison between these two models. While reading these two modeling approaches, the reader should maintain this perspective and treat them separately.
Our key contributions are:

- We propose a novel model CHARSPAN [MKDK24]: *Character-Span Noise Augmentation*, which considers surface-level lexical similarity to improve cross-lingual transfer between closely-related HRLs and ELRLs. The proposed approach shows, on average, 12.5% chrF improvement over baseline NMT models across all considered ELRLs.

- We propose a novel SELECTNOISE[4] [BMD23]: *Unsupervised Selective Character Noise Augmentation* approach to improving cross-lingual transfer between closely-related HRLs and ELRLs. This is an unsupervised mechanism to *extract*

[4] Equal contribution with my co-author Maharaj Brahma

selective candidate characters based on BPE merge operations and edit operations. Furthermore, the *noise augmentation* employs greedy, top-k, and top-p sampling techniques to sample noise augmentation candidates. The proposed approach achieved a cumulative gain of 11.3% chrF over baseline NMT models.

- The zero-shot evaluations are conducted with both automated and human evaluation metrics. The evaluations are done across several ELRLs from typologically diverse language families.
- We have conducted extensive ablation and analyses for both models to gain insights and demonstrate the effectiveness of the proposed approaches.

7.2 Related Work

In this section, we review three threads of literature related to the proposed models: (1) MT for LRLs/dialects, (2) vocabulary adaptation for low-resource MT, and (3) data augmentation for low-resource MT.

7.2.1 MT for Low-resource Languages/Dialects

Due to the unavailability of the large bi-text dataset for LRLs, much of the existing research focuses on *multilingual* MT. This enables cross-lingual transfer [NC17, ZYMK16] and allows related languages to learn from each other [FBS+21, CjCÇ+22, SBF+22]. While this direction has gained significant attention, the performance improvement for LRLs as compared to HRLs has been limited [TBC+21] and remains an open area of research. In another thread, efforts have been made for low-resource MT models directly from the monolingual dataset [ALAC18, LCDR18, LLG+20b]. These unsupervised approaches show promise but still require a large amount of monolingual data, which should ideally match the domain of the HRLs [MDK20]. However, for many LRLs, monolingual datasets are often unavailable [ARY+20b]. In contrast, we propose models that do not require any parallel data and no monolingual data (CharSpan)/ limited monolingual datasets (SelectNoise). This characteristic ensures the scalability of our proposed models to many ELRLs.

7.2.2 Vocabulary Adaptation for Low-resource MT

Early exploration of character-based MT showed the promise [CCB16, LCH17] with coverage and robustness [PEV20, LF20]. However, recent modeling concludes a number of challenges [GBDG19, LF20] in terms of training/inference times and performance as compared to the subwords models. Specifically, [SL21] shows that character MT and Byte MT [CjEF17] have worse performance than the Byte Pair Encoding (BPE; [SHB16b]) model and have limited practical usage [LSF22]. The effectiveness of the BPE algorithm [Gag94] is reported for NMT [SHB16b] and serval other NLP tasks [LOG+19]. Other algorithms like Sentencepiece [KR18] and Wordpiece [Goo] are similar to BPE. We take inspiration from existing works and proposed a model based on BPE.

Given the potential of the BPE model, various methodologies have been developed for vocabulary modification/generation/adaption [PEV20, KMP+21, PTS22, MPR22]. In particular, the work of [PEV20] utilizes the BPE algorithm to generate the vocabulary and sample different segmentations during training. [PTS22] introduce an extension of BPE, referred to as Overlapped BPE (OBPE), which takes into account both HRLs and LRLs tokens during vocabulary creation. They demonstrate the effectiveness of this approach in only NLU tasks. In contrast, in this study, we adapt the standard BPE model to learn vocabulary with noisy HRL data for an NLG task, i.e., MT. The proposed noise augmentation-based modeling effectively learns a vocabulary that improves cross-lingual transfer from HRLs to LRLs.

7.2.3 Data Augmentation for Low-resource MT

The limited availability of parallel data leads to a wide range of data augmentation approaches [ZWL+19, GZW+19, CK18]. Traditional approaches include perturbation at the word level, such as word dropout [SHB16a], word replacement [WPDN18] and soft-decoupling (SDE; [WPAN19a]) to improve the cross-lingual transfer for LRLs. Such perturbation acts as a regularizer and enhances robustness to spelling variations; however, their impact is limited [AS22]. In a different research direction, noise augmentation-based modeling [SNW17, KLEG19] has been explored to test the robustness of MT systems. More recently, lexical match-based models have been explored to improve the cross-lingual transfer by vocabulary overlapping [PTS22], non-deterministic segmentation [PEV20] and noise augmentation [AS22, BSP23]. Noise augmentation models are close to our proposed models. However, these models have been evaluated with only NLU tasks using pre-trained models. In contrast, we have trained our model from scratch specifically for the MT task, which is more challenging than NLU tasks. Additionally, we introduce two novel approaches: character-span noise and linguistically inspired systematic noise augmentation techniques tailored for ELRLs.

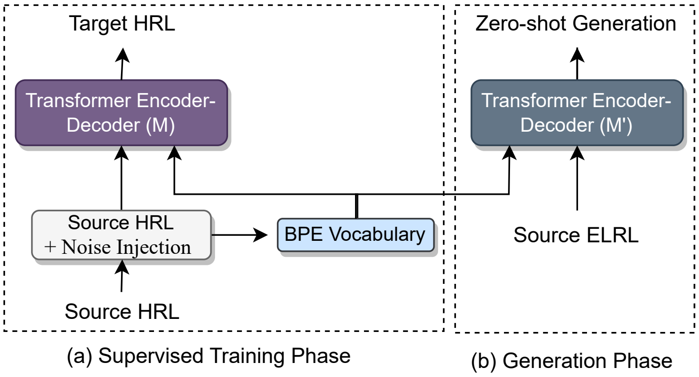

Target HRL
Transformer Encoder-Decoder (M)
Source HRL + Noise Injection
BPE Vocabulary
Source HRL
(a) Supervised Training Phase
Zero-shot Generation
Transformer Encoder-Decoder (M')
Source ELRL
(b) Generation Phase

Figure 7.3: Overview of proposed CharSpan model

7.3 Character Span Noise Augmentation

7.3.1 Methodology

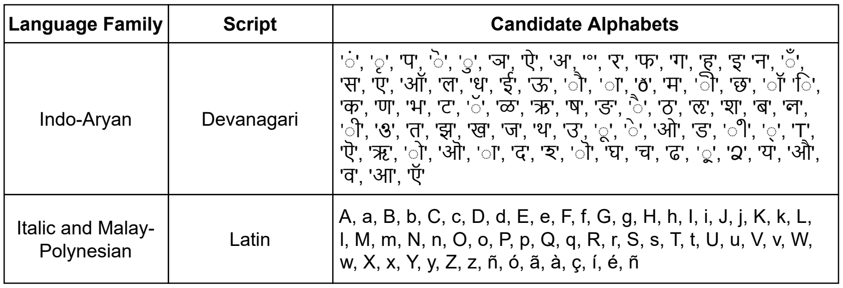
In this section, we present details of the first proposed model: CharSpan. Figure 7.3 illustrates an overview of the proposed CharSpan model for the ELRLs to English MT task. The model has two phases: supervised training with noisy HRL and zero-shot generation with ELRLs.

Model Training and Generation:

In the *supervised training phase*, the source-side training data of the HRL pair ($\mathcal{D}_{\mathcal{H}}$) is augmented with character-span noise (described later) to create the augmented parallel corpus ($\mathcal{D}'_{\mathcal{H}} = \eta(\mathcal{D}_{\mathcal{H}})$), where η is the noise function. $\eta(\mathcal{D}_{\mathcal{H}})$ can be considered as the proxy parallel data for the ELRL-English translation task. Next, we learn a subword vocabulary ($\mathcal{V}$) using $\mathcal{D}'_{\mathcal{H}}$, i.e., the noise is augmented before learning the vocabulary. A standard encoder-decoder transformer model ($\mathcal{M}$; [VSP+17]) is then trained with $\mathcal{D}'_{\mathcal{H}}$ and $\mathcal{V}$ from scratch in a supervised setting to obtain the trained model $\mathcal{M}'$. Finally, in the *zero-shot generation phase*, for a given source ELR language $\mathcal{L}$, the target English translation is obtained using $\mathcal{M}'$ and $\mathcal{V}$ in the zero-shot setting.

Character-Span Noise (CSN) Function:

The character-span noise function makes the model robust to spelling variations between related languages. This acts as a regularizer and helps improve cross-lingual representation and transfer. Intuitively, the existing unigram character noise [AS22] might address limited lexical variations between HRL and ELRLs. *To address larger lexical divergence, we propose a CSN where span noise is augmented.* Formally, for a given sentence, $x \in \mathcal{X}$ from $\mathcal{D}'_{\mathcal{H}}(\mathcal{X}, \mathcal{Y})$ with indices $I = 1, 2, \ldots, |x|$, a subset of these indices $I_s \subset I$ is randomly and uniformly selected as the starting point for the noise augmentation. Subsequently, 1-3 character gram spans are iteratively sampled until the noise augmentation budget (i.e., 9% - 11% characters) is exhausted. We employ *span deletion* and *span replacement with a single random character of ELRL*, both with equal probability as the noising operations. This CSN is inspired by SpanBERT [JCL+20][5]. Sample candidate characters for noise augmentation (for replacement operation) are shown in Fig. 7.4. A formal algorithm is presented in the Algorithm 2. We conducted experiments with all three operations (including insertion), different percentages of noise, and various other experimental setups. We found the presented noise augmentation configuration is the most effective.



Language Family	Script	Candidate Alphabets
Indo-Aryan	Devanagari	'ं', 'ृ', 'प', 'े', 'ु', 'ञ', 'ऐ', 'अ', '॰', 'र', 'फ', 'ग', 'ह', 'इ' 'न', 'ँ', 'स', 'ए', 'ऑ', 'ल', 'ध', 'ई', 'ऊ', 'ौ', 'ा', 'ठ', 'म', 'ी', 'छ', 'ॉ' 'ि', 'क', 'ण', 'भ', 'ट', 'ॅ', 'ळ', 'ऋ', 'ष', 'ङ', 'ै', 'ठ', 'ऌ', 'श', 'ब', 'ऴ', 'ी', '७', 'त', 'झ', 'ख', 'ज', 'थ', 'उ', 'ू', 'ॆ', 'ओ', 'ड', 'ी', '्', 'T', 'ऐ', 'ॠ', 'ो', 'ओ', 'ा', 'द', 'ऱ', 'ो', 'घ', 'च', 'ढ', 'ू', '२', 'य', 'औ', 'व', 'आ', 'ऍ'
Italic and Malay-Polynesian	Latin	A, a, B, b, C, c, D, d, E, e, F, f, G, g, H, h, I, i, J, j, K, k, L, l, M, m, N, n, O, o, P, p, Q, q, R, r, S, s, T, t, U, u, V, v, W, w, X, x, Y, y, Z, z, ñ, ó, ã, à, ç, í, é, ñ

Figure 7.4: Candidate alphabets for noise augmentation from ELRLs

7.3.2 Experimental Setup

We seek answers to the following questions: (1) Does the augmentation of CSN improve cross-lingual transfer, i.e., zero-shot performance for related ELRLs for MT task? (2) Why does the model's cross-lingual transfer improve? - Insights from the learned embedding space. (3) Is the proposed approach scalable to typologically diverse language families? (4) Is the model's generalization ability maintained when

[5] SpanBERT applies denoising to subword tokens while we apply it at the character level.

Algorithm 2 CharSpan: Character-span Noise Augmentation Algorithm

```
Require: [Inputs] high resource language data ($\mathcal{D}_{\mathcal{H}}(\mathcal{X}, \mathcal{Y})$) from $H$-$En$ parallel corpus, range of noise augmentation percentage $[P1, P2]$, set of noise augmentation candidates $C$ (see Fig. 7.4), largest character $n$-gram size $N$ that will be considered for noising
Ensure: [Output] Noisy high resource language data ($\mathcal{D}^{'}_{\mathcal{H}}$)
Augmentation percentage ($I_p$) = random float(P1, P2)    ▷ Find a random float value between $P1$ and $P2$
Augmentation factor ($\alpha$) = int($I_p/N$)
for each $h$ in $\mathcal{X}$ do
    Let $sz$ be the number of characters in $h$.
    Let $Indices = \{\lceil(N/2)\rceil, \cdots, sz - \lceil(N/2)\rceil\}$    ▷ Leaving $\lceil(N/2)\rceil$ character indices from beginning and end
    Randomly select $S = N * \alpha$ character indices from $Indices$
    for each $k$ in $S$ do
        Span gram ($Sp_N$) = sample character-span size uniformly from $\{1, 2, \ldots, N\}$ with equal probability
        Operation ($O_p$) = sample operations uniformly from { delete, replace } with equal probability
        $C_d$ ={}
        if ($O_p$) is replace then
            Candidate char ($c$) = single sample character uniformly from $C$ with equal probability
            Append candidate char $c$ in $C_d$
        end if
        if $Sp_N == 1$ then
            Perform the operation ($O_p$) with $C_d$ at the index $k$
        else
            Perform the operation ($O_p$) with $C_d$ at the indexes from $k - \lceil((Sp_N - 1)/2)\rceil$ to $k + \lceil((Sp_N - 1)/2)\rceil$
        end if
    end for
end for
```

using smaller parallel training data of HRLs? Considering these, we have designed the following experimental setup:

Datasets and Languages

We evaluated the performance of the CharSpan on three language families: Indo-Aryan, Romance, and Malay-Polynesian. We considered six HRLs and twelve LRLs (two HRLs and several ELRLs from each family). All the ELRLs are lexically similar and have the same script with corresponding HRLs. Parallel training data for the HRLs was selected from publicly available datasets. The model's performance was evaluated on the FLORES-200 devtest set [CjCÇ+22]. All the dataset details are

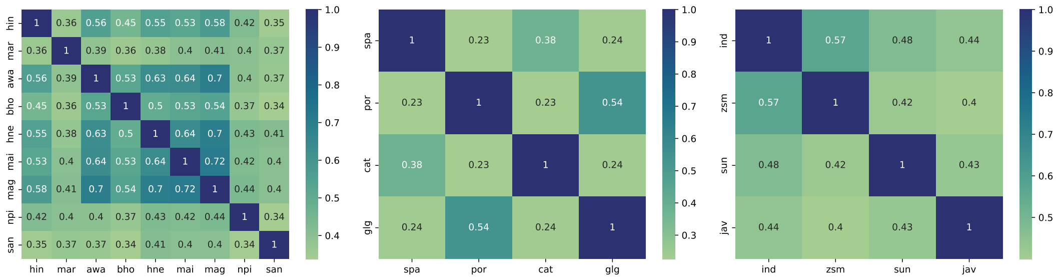


Figure 7.5: Heatmaps showing lexical similarity (LCSR) are presented for three language families. The Indo-Aryan languages have the Devanagari script, whereas languages from the Romance and Malay-Polynesian families have the Latin script.

presented in Table 7.1. Further, Fig. 7.5 presents a lexical similarity heatmap for the considered language families and ELRLs. It can observed that ELRLs are closely related to corresponding HRLs. The lexical similarity between languages was measured using character-level longest common sub-sequence ratio (LCSR) metric [Mel95].

Family	Code	Language	Script	Family	Subgrouping	Res.	Train	Dev	Test	Data Source
1	Hin	Hindi	Devanagari	Indo-European	Indo-Aryan	High	10M	1000	2390	[RDB+22]
	Mar	Marathi	Devanagari	Indo-European	Indo-Aryan	High	3.6M	1000	2390	[RDB+22]
	Bho	Bhojpuri	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	Gom	Konkani	Devanagari	Indo-European	Indo-Aryan	Low	-	-	2000	ILCI[6]
	Hne	Chhattisgarhi	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	San	Sanskrit	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	Npi	Nepali	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	Mai	Maithili	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	Mag	Magahi	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
	Awa	Awadhi	Devanagari	Indo-European	Indo-Aryan	Low	-	-	1012	FLORES-200
2	Spa	Spanish	Latin	Indo-European	Romance	High	6.6M	670	1131	[Rap21]
	Pot	Portuguese	Latin	Indo-European	Romance	High	4.8M	681	1103	[Rap21]
	Cat	Catalan	Latin	Indo-European	Romance	Low	-	-	1012	FLORES-200
	Glg	Galician	Latin	Indo-European	Romance	Low	-	-	1012	FLORES-200
3	Ind	Indonesian	Latin	Austronesian	Malay-Polynesian	High	0.5M	2500	3000	OPUS[7]
	Zsm	Malay	Latin	Austronesian	Malay-Polynesian	High	0.3M	1500	2000	OPUS
	Jav	Javanese	Latin	Austronesian	Malay-Polynesian	Low	-	-	1012	FLORES-200
	Sun	Sundanese	Latin	Austronesian	Malay-Polynesian	High	-	-	1012	FLORES-200
Others	Pan	Panjabi	Gurmukhi	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200
	Guj	Gujarati	Gujarati	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200
	Ben	Bengali	Bengali	Indo-European	Indo-Aryan	High	1M*	1000*	1012	FLORES-200
	Tam	Tamil	Tamil	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200
	Tel	Telugu	Dravidian	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200
	Mal	Malayalam	Malayalam	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200
	Ora	Oriya	Oriya	Indo-European	Indo-Aryan	Low	1M*	1000*	1012	FLORES-200

Table 7.1: Dataset details and Statistics. * are obtained from [RDB+22]

Baseline Models

We compare the proposed model performance with the following strong baselines:

- **Vanilla NMT (BPE; [SHB16b]):** Neural Machine Translation model training with the standard BPE algorithm.

- **WordDropout [SHB16a]:** In this baseline, the embeddings of randomly selected 10% words in the source sentences of HRL to 0.

- **SubwordDropout:** It is a variant of WordDropout baseline where we drop the BPE tokens instead of words.

- **WordSwitchOut [WPDN18]:** This baseline employs a data augmentation technique where random words in both the source and target sentences are replaced with randomly selected words from their respective vocabularies. We have utilized the officially released implementation with a 10% word replacement rate.

- **SubwordSwitchOut:** It is a variant of WordSwitchOut baseline where we use the BPE tokens instead of words.

- **Overlap BPE (OBPE; [PTS22]):** The approach modifies the BPE algorithm to encourage more shared tokens from HRLs and LRLs in the vocabulary. This model required a monolingual dataset for LRLs. We use a small monolingual dataset, based on availability, for the ELRLs. Earlier work applied OBPE for NLU tasks only - we are the first to investigate it for MT.

- **Soft Decoupled Encoding (SDE; [WPAN19b]):** In the SDE approach, the authors have designed a framework that effectively decouples word embeddings into two interacting components: representing the spelling of words and capturing the latent meaning of words. This modeling technique has demonstrated its effectiveness in improving the performance of LRLs. We use the officially released implementation of SDE.

- **BPE-Dropout [PEV20]:** It utilizes the BPE algorithm to learn the vocabulary and sample different segmentations for input text during training (on-the-fly).

- **Unigram Character Noise (UCN; [AS22]):** This model augments character-level noise (with all three operations) unlike CharSpan where we augment span level noise (with only two operations).

- **BPE → Char-Span Noise:** In this ablation study, we learn the vocabulary using clean HRLs training data. Following that, we introduce character-span noise into the source side of HRLs. This helps us understand whether learning a BPE vocabulary is effective in which scenario, with or without noisy HRL training data.

- **Char-Span Noise + BPE-Dropout:** In this model, we train the BPE-Dropout model with char-span noise augmented HRLs training dataset.

Evaluation Metrics

In line with recent studies on MT for ELRLs [CjCÇ+22, SBF+22], we use chrF[8] [Pop15], and BLEU[9] [PRWZ02d] are lexical overlap based metrics. Further, two learned neural metrics viz., BLEURT [SDP20] and COMET [RSFL20] are used.

Implementation Details

We used the FairSeq library [OEB+19] to train proposed CharSpan and other baseline models from scratch. The different hyper-parameter details are presented in Table 7.2. The best checkpoint was selected based on validation loss. The *CharSpan* model training time for the Indo-Aryan family was approximately 8 hours; for the Romance languages it was approximately 7 hours, and for the Malay-Polynesian, it was less than 1 hour. For each ELRL, the zero-shot generation time was less than 5 minutes. Due to computational limitations, the performance of the model was reported based on a single run. During the generation process, a batch size of 64 and a beam size of 5 were used—the rest of the model parameters were set to the default values as per FairSeq. For data-pre-processing and script conversion (for Indic languages) we use the Indic NLP library[10]

7.3.3 Results and Analyses

The automated evaluation results for the CharSpan and baseline models across all language families are presented in Tables 7.3, 7.4, 7.5 and 7.6. The following are the major observations:

Noise vs. Baselines: All the proposed noise augmentation models outperform vanilla NMT and all baseline models that utilize lexical similarity (i.e., OBPE, BPE-Dropout, and SDE). This trend is consistent across all language families and ELRLs. Moreover, existing lexical similarity-based baselines do not provide

[8] SacreBLEU chrF signature: *nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1*

[9] SacreBLEU BLEU signature: *nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1*

[10] https://github.com/anoopkunchukuttan/indic_nlp_library

Criteria	**Associated Values**
architecture	encoder-decoder (transformers)
Number of encoder layers	6
Number of decoder layers	6
Number of parameters	46,956,544 shared
learning rate (lr)	$5e^{-4}$
optimizer	adam
dropout rate	0.2
input size	210 tokens (both side)
epochs	15
tokens per batch	32768
clip-norm	1.0
learning rate scheduler	inverse sqrt
Number of GPUs	8
type of GPU	V100 Nvidia
generation batch size	64
beam size	5

Table 7.2: Model implementation and training details

Models	**Indo-Aryan**								**Romance**		**Malay-Polynesian**		**Average**
	Gom	**Bho**	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**	**Cat**	**Glg**	**Jav**	**Sun**	
BPE	26.75	39.75	46.57	27.97	30.84	39.79	48.08	46.28	33.32	53.75	31.44	32.21	38.06
WordDropout	27.01	39.57	46.19	28.13	31.91	40.31	47.37	46.48	34.20	52.21	32.03	32.52	38.16
SubwordDropout	27.91	40.11	46.26	29.46	32.56	40.99	47.91	47.43	35.09	52.28	33.38	33.47	38.90
WordSwitchOut	25.17	38.81	45.87	26.21	29.95	39.69	47.53	44.54	32.98	51.81	31.84	32.49	37.24
SubwordSwitchOut	26.08	38.84	45.84	28.19	30.81	40.19	47.28	45.93	33.26	53.71	31.24	32.06	37.78
OBPE	27.90	40.57	47.46	28.52	31.99	40.71	49.10	47.16	32.33	52.77	29.98	30.88	38.28
SDE	28.01	40.91	47.88	28.66	32.03	40.82	48.96	47.30	33.72	53.95	31.84	31.24	38.77
BPE-Dropout	28.65	40.84	46.58	28.80	31.88	40.79	47.86	47.32	34.56	55.83	32.01	32.97	39.00
unigram char-noise	28.85	42.53	49.35	29.80	34.61	42.67	50.97	49.43	43.16	54.81	35.42	36.69	41.52
BPE → CSN (*our*)	28.66	41.94	49.48	30.49	35.66	44.75	50.55	49.21	43.11	54.89	36.12	37.11	40.16
CharSpan (*our*)	29.71	43.75	51.69	**31.40**	36.52	45.84	51.90	50.55	43.51	55.46	36.24	37.31	42.82
CharSpan + BPE-Dropout (*our*)	**29.91**	**44.02**	**51.86**	30.88	**37.15**	**46.52**	**52.99**	**51.34**	**44.93**	**55.87**	**36.97**	**38.09**	**43.37**

Table 7.3: Zero-shot chrF scores results for ELRLs → English MT.

Models	**Indo-Aryan**								**Romance**		**Malay-Polynesian**		**Average**
	Gom	**Bho**	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**	**Cat**	**Glg**	**Jav**	**Sun**	
BPE	4.36	10.62	15.76	3.43	4.36	9.36	16.7	15.6	5.23	22.99	5.74	6.02	10.01
WordDropout	4.62	11.21	15.71	4.11	5.47	9.96	16.76	16.31	6.19	22.26	5.90	6.02	10.37
SubwordDropout	4.57	9.99	14.47	3.93	5.25	9.08	15.53	16.03	5.85	20.72	4.78	4.93	09.59
WordSwitchOut	4.03	10.75	15.86	3.56	4.92	9.91	16.85	15.54	5.27	21.97	5.95	6.35	10.08
SubwordSwitchOut	4.13	10.56	15.93	3.76	4.49	9.69	16.61	16.69	5.19	23.82	6.02	6.01	10.24
OBPE	4.65	10.62	16.31	3.63	4.95	9.18	16.88	15.69	5.03	22.91	5.33	5.81	10.08
SDE	4.77	10.69	16.21	3.66	5.42	9.86	16.80	16.03	5.47	23.51	5.88	6.39	10.39
BPE-Dropout	5.24	11.33	15.64	3.71	4.94	10.00	16.62	16.63	5.94	24.07	5.79	6.65	10.54
unigram char-noise	5.21	12.62	18.29	3.81	6.55	11.29	19.47	18.95	11.82	24.09	7.35	6.87	12.19
BPE → CSN (*our*)	5.39	13.06	19.00	4.48	7.01	13.17	20.30	19.69	11.91	24.27	7.51	7.30	12.75
CharSpan (*our*)	5.77	13.01	19.52	4.63	7.13	13.43	20.81	20.36	12.21	24.72	7.52	7.32	13.03
CharSpan + BPE-Dropout (*our*)	**5.81**	**13.81**	**21.03**	**4.64**	**8.10**	**14.33**	**22.11**	**21.25**	**12.64**	**25.35**	**7.52**	**7.31**	**13.65**

Table 7.4: Zero-shot BLEU scores results for ELRLs → English MT

any major improvement in translation quality over vanilla NMT. Possible reasons for this can be two-fold: (1) most of the ELRLs either do not have monolingual data or have small data (OBPE and SDE require large monolingual data), and (2) we observe that in OBPE, approximately 90% of vocabulary tokens are

Models	Indo-Aryan								Romance		Malay-Polynesian		Average
	Gom	**Bho**	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**	**Cat**	**Glg**	**Jav**	**Sun**	
BPE	0.461	0.494	0.522	0.414	0.461	0.494	0.537	0.549	0.357	0.495	0.403	0.401	0.474
WordDropout	0.467	0.502	0.527	0.419	0.465	0.497	0.542	0.565	0.344	0.496	0.392	0.391	0.475
SubwordDropout	0.454	0.493	0.513	0.393	0.459	0.481	0.526	0.554	0.319	0.468	0.382	0.383	0.460
WordSwitchOut	0.456	0.501	0.528	0.395	0.445	0.497	0.552	0.551	0.309	0.477	0.381	0.381	0.464
SubwordSwitchOut	0.459	0.494	0.519	0.415	0.455	0.496	0.535	0.555	0.365	0.496	0.383	0.385	0.467
OBPE	0.466	0.496	0.518	0.419	0.459	0.491	0.537	0.551	0.431	0.428	0.396	0.381	0.464
SDE	0.486	0.499	0.515	0.511	0.496	0.542	0.543	0.553	0.440	0.481	0.406	0.405	0.489
BPE-Dropout	0.474	0.494	0.501	0.413	0.461	0.481	0.522	0.555	0.443	0.443	0.407	0.412	0.467
unigram char-noise	0.471	0.523	0.547	0.403	0.456	0.486	0.571	0.592	0.495	0.501	0.403	0.405	0.487
BPE → CSN (*our*)	0.469	0.528	0.553	0.400	0.459	0.491	0.579	0.595	0.499	0.511	0.405	0.413	0.491
CharSpan (*our*)	0.471	0.541	0.571	0.403	0.471	0.534	0.593	0.616	0.502	0.555	**0.419**	0.422	0.508
CharSpan + BPE-Dropout (*our*)	**0.478**	**0.548**	**0.582**	**0.421**	**0.478**	**0.535**	**0.604**	**0.623**	**0.505**	**0.567**	**0.419**	**0.429**	**0.515**

Table 7.5: Zero-shot BLEURT (computed with *BLEURT-20* checkpoint) scores results for ELRLs → English MT

Models	Indo-Aryan								Romance		Malay-Polynesian		Average
	Gom	**Bho**	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**	**Cat**	**Glg**	**Jav**	**Sun**	
BPE	0.536	0.632	0.671	0.511	0.525	0.593	0.694	0.716	0.494	0.714	0.444	0.441	0.580
WordDropout	0.551	0.648	0.678	0.521	0.557	0.618	0.695	0.728	0.565	0.715	0.451	0.443	0.597
SubwordDropout	0.541	0.638	0.659	0.528	0.548	0.607	0.684	0.717	0.524	0.686	0.437	0.428	0.583
WordSwitchOut	0.544	0.647	0.681	0.522	0.563	0.621	0.706	0.719	0.529	0.702	0.453	0.452	0.594
SubwordSwitchOut	0.542	0.641	0.668	0.521	0.528	0.601	0.694	0.721	0.567	0.718	0.452	0.451	0.592
OBPE	0.541	0.629	0.667	0.504	0.527	0.589	0.691	0.715	0.492	0.721	0.363	0.611	0.587
SDE	0.549	0.636	0.666	0.513	0.529	0.591	0.697	0.735	0.513	0.731	0.357	0.618	0.594
BPE-Dropout	0.549	0.638	0.644	0.506	0.531	0.589	0.677	0.721	0.504	0.747	0.373	0.626	0.592
unigram char-noise	0.562	0.679	0.701	0.536	0.573	0.634	0.728	0.754	0.554	0.741	0.408	0.621	0.624
BPE → CSN (*our*)	0.557	0.676	0.706	0.542	0.581	0.651	0.724	0.755	0.561	0.751	0.403	0.622	0.627
CharSpan (*our*)	0.571	0.695	0.723	**0.556**	0.611	0.685	0.747	0.772	0.568	**0.759**	**0.417**	0.627	0.644
CharSpan + BPE-Dropout (*our*)	**0.579**	**0.705**	**0.733**	0.551	**0.616**	**0.687**	**0.757**	**0.778**	**0.572**	0.756	0.414	**0.631**	**0.648**

Table 7.6: Zero-shot COMET (computed with *Unbabel/wmt22-comet-da* model) scores results for ELRLs → English MT

already overlapping among HRLs and ELRLs, leaving little room for learning additional overlapping tokens. This is expected, as HRLs and LRLs are closely related. The proposed CharSpan method also outperforms general data augmentation methods like (Sub)WordDropout and (Sub)WordSwitchout, showing its effectiveness.

Unigram vs. Char-Span Noise: We are first to explore unigram char noise [AS22] for MT task. We see that unigram char noise is beneficial for the task. However, our proposed CharSpan provides significant improvements over unigram character noise. We believe our proposed data augmentation is more effective in bringing language representations closer. It also offers performance gains for languages like Konkani (Gom), which are distantly similar to the HRLs as other languages.

When to introduce noise? To understand when noise augmentation is effective, we augmented noise after learning the vocabulary in the baseline (BPE → CSN). This leads to improved performance over all baselines. This enables scalability since augmenting noise after learning the vocabulary allows the applicability of this method to large language models that have fixed vocabulary. However, the results suggest that applying noise prior to learning the vocabulary, as in CharSpan, yields slightly better results.

Combining noise and BPE-dropout: We see that combining CSN with BPE-dropout gives the best-performing results.

Langs.	**BPE**	**Unigram Noise**	**CharSpan**	**Sim**
Guj-Deva	34.36	36.17	38.09	0.42
Pan-Deva	29.18	33.34	36.50	0.40
Ben-Deva	25.35	28.42	30.28	0.34
Tel-Deva	23.30	24.05	24.12	0.27
Tam-Deva	13.81	13.69	14.40	0.15

Table 7.7: Zero-shot chrF scores with additional lexically less similar languages. `HRL:` hi and mr; `sim:` lexical similarity

Performance on Less Similar Languages: We evaluate the *CharSpan* model's performance on languages that are less lexically similar to the considered HRLs and have different scripts. The languages are Gujarati (Guj), Punjabi (Pan), Bengali (Ben), Telugu (Tel), and Tamil (Tam). We perform script-conversion [Kun20]) of these languages to HRL script. The training and evaluation setup is similar to the Indo-Aryan family. Table 7.7 shows that the ELRLs, which are lexically similar to HRLs, demonstrate a larger performance gain, while those with less lexical similarity show limited improvement. This suggests that the model's effectiveness is closely tied to the lexical similarity of the languages. The lexical similarity scores for these less similar languages are illustrated in Fig 7.6.

Impact of Cross-lingual Transfer: In this analysis, we investigate the encoded representations of the sentences to gain insights into how performance improves with char-span noise augmentation. We collected pooled last-layer representations of the encoder for HRL and LRLs across all parallel test examples using BPE, unigram char-noise (UCN), and the CharSpan models. We then calculated the average cosine similarity scores across the test set, presented in Table 7.8. Notably, the CharSpan model demonstrates high similarity, indicating a well-aligned embedding space that enhanced cross-lingual transfer.

Importance of Selecting Right HRLs: Table 7.9 presents an analysis of the impact of lexically diverse HRLs used for training. Results indicate that the CharSpan model demonstrates a performance gain when lexically similar HRLs

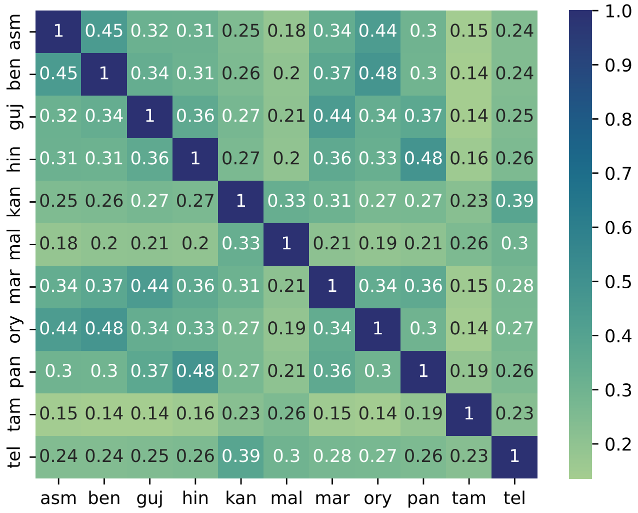


Figure 7.6: Lexical similarity heatmap for additional languages used in the analysis section. Here we have shown similarity scores for Assamese (asm), Bengali (ben), Gujrati (guj), Panjabi (pan), Hindi (him), Marathi (mar), Oriya (ory), Malayalam (mal), Kannada (kan), Tamil (tam) and Telugu (tel) languages.

Models	**Indo-Aryan**							**Romance**		**Malay-Polynesian**		**Average**
	Bho	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**	**Cat**	**Glg**	**Jav**	**Sun**	
BPE	0.761	0.793	0.701	0.744	0.762	0.809	0.792	0.721	0.813	0.731	0.736	0.760
UCN	0.853	0.888	0.765	0.821	0.849	0.897	0.883	0.803	0.879	0.813	0.811	0.842
CharSpan	**0.871**	**0.909**	**0.789**	**0.858**	**0.868**	**0.913**	**0.901**	**0.831**	**0.903**	**0.846**	**0.856**	**0.867**

Table 7.8: Average cosine similarity between representations of source HRLs and source ELRLs. UNC: Unigram Char-Noise

were considered for noise augmentation. When the HRLs are less lexically similar, a degradation in performance is observed. These findings indicate the importance of using lexically similar HRLs. The lexical similarity scores for these additional languages are illustrated in Fig 7.6.

Model	**Hne**	**Mag**	**Mai**	**Npi**	**San**
Training with Lexically Similar HRLs: Hin, Mar, Pan, Guj, Ben					
BPE	43.04	45.08	39.51	31.92	29.29
Char-span Noise	45.89	45.82	41.67	34.40	30.34
Training with Lexically less similar HRLs: Hin, Tel, Tam, Mal, Ora					
BPE	41.87	42.27	36.95	30.50	26.95
Char-span Noise	39.93	40.34	37.98	29.20	25.84

Table 7.9: Analysis experiment to show zero-shot chrF scores with lexically diverse HRLs. Due to computational constraints, we have considered 1 million parallel data for each HRL.

Impact of Small ELRL Parallel Data: Here, we combined small ELRL parallel data with the HRLs training data for BPE and CharSpan model. The results are presented in Table 7.10. The additional data boosts both model performances. However, CharSpan still outperforms the BPE model.

Impact of Less HRL Parallel Data: For the Malay-Polynesian family, we use only approximately 10% of the HRL parallel training data compared to the other two language families. However, it can be observed that despite having limited training data, the CharSpan model outperforms all the baselines and exhibits similar performance trends as the other two language families. This conclusion suggests that the proposed model remains effective even with a small amount of HRL training data.

Performance with Different Automated Evaluation Metrics: To ensure the performance gains are genuine, we have evaluated the CharSpan performance using four automated evaluation metrics. It can be observed that similar performance trends are evident for all metrics and are correlated with each other. This indicates the reliability of the experimental evaluation.

Setup	**Gom**	**Bho**	**Hne**	**San**	**Npi**	**Mai**
BPE	26.75	39.75	46.57	27.97	30.84	39.79
BPE+ELRL$_{\text{par}}$	26.54	42.66	52.52	31.88	38.09	43.22
CSN	29.71	43.75	51.69	31.40	36.52	45.84
CSN+ELRL$_{\text{par}}$	29.65	45.39	53.38	33.92	39.66	47.18

Table 7.10: Translation quality (chrF) with an additional 1000 ELRL-English parallel sentences (ELRL$_{\text{par}}$).

7.3.4 Further Analyses and Discussions

Performance on High Resource Languages: The high-resource language performances are presented in Table 7.11. It can be observed that, even with the inclusion of noise augmentation, the proposed model exhibits only a slight decrease in performance for HRLs.

Ablation Study and Different Experimental Setups: In order to ascertain the optimal configuration of the proposed model, a comprehensive set of experiments, numbering approximately 200, were conducted. Selected key experiments are

XX → EN	Indo-Aryan				Romance				Malay-Polynesian			
Models	**BLEU**		**chrF**		**BLEU**		**chrF**		**BLEU**		**chrF**	
	Hin	**Mar**	**Hin**	**Mar**	**Spa**	**Pot**	**Spa**	**Pot**	**Ind**	**Zsm**	**Ind**	**Zsm**
BPE	37.44	26.31	64.04	54.47	41.44	35.38	68.71	63.27	29.61	21.76	58.31	49.14
WordDropout	36.54	26.31	63.27	53.96	39.32	32.73	66.89	60.86	27.59	20.42	56.72	48.22
SubwordDropout	36.64	26.22	63.46	54.57	39.84	33.04	67.56	61.58	26.73	18.80	57.02	48.82
WordSwitchOut	34.12	23.84	60.98	51.84	35.27	30.63	63.25	58.38	27.04	19.60	55.69	46.93
SubwordSwitchOut	37.11	26.03	63.78	54.06	42.26	35.68	68.65	62.97	27.12	19.76	55.72	47.34
OBPE	37.32	26.90	64.05	55.03	41.81	36.44	68.17	63.45	28.14	21.83	57.11	49.21
SDE	37.22	26.19	63.98	55.44	41.41	35.51	68.61	62.89	29.11	21.52	58.25	48.98
BPE-Dropout	37.22	26.93	64.11	55.31	41.88	36.72	68.06	63.79	30.39	22.54	59.33	50.17
unigram char-noise	37.05	26.95	63.81	54.83	39.83	32.91	67.62	61.24	28.79	22.01	57.65	49.91
BPE → CSN (*our*)	36.66	26.93	63.80	54.84	39.92	32.22	66.83	61.06	27.84	22.16	57.15	50.19
CharSpan (*our*)	36.68	26.70	63.87	54.59	40.04	32.36	66.95	61.03	27.84	21.87	56.75	49.58
CharSpan + BPE-Dropout (*our*)	37.62	27.10	64.15	55.03	41.21	33.64	66.90	61.39	28.91	22.26	57.99	50.59

Table 7.11: BLEU and chrF Scores: HRLs performance for all three language families

XX → EN	Indo-Aryan				Romance				Malay-Polynesian			
Models	**BLEURT**		**COMET**		**BLEURT**		**COMET**		**BLEURT**		**COMET**	
	Hin	**Mar**	**Hin**	**Mar**	**Spa**	**Pot**	**Spa**	**Pot**	**Ind**	**Zsm**	**Ind**	**Zsm**
BPE	0.775	0.726	0.891	0.857	0.769	0.720	0.871	0.830	0.687	0.561	0.821	0.701
WordDropout	0.774	0.725	0.891	0.854	0.755	0.701	0.86	0.814	0.681	0.555	0.815	0.693
SubwordDropout	0.773	0.725	0.889	0.854	0.757	0.691	0.861	0.806	0.672	0.548	0.803	0.683
WordSwitchOut	0.756	0.706	0.879	0.842	0.707	0.651	0.826	0.775	0.665	0.547	0.804	0.688
SubwordSwitchOut	0.776	0.724	0.892	0.855	0.771	0.721	0.872	0.833	0.663	0.548	0.801	0.687
OBPE	0.777	0.731	0.893	0.861	0.766	0.727	0.863	0.821	0.672	0.551	0.811	0.697
SDE	0.772	0.721	0.889	0.856	0.765	0.721	0.866	0.832	0.679	0.558	0.818	0.699
BPE-Dropout	0.773	0.727	0.891	0.858	0.772	0.7281	0.881	0.839	0.706	0.586	0.838	0.729
unigram char-noise	0.775	0.731	0.892	0.857	0.756	0.683	0.861	0.798	0.681	0.574	0.815	0.716
BPE → CSN (*our*)	0.773	0.728	0.891	0.857	0.755	0.685	0.861	0.801	0.685	0.581	0.821	0.724
CharSpan (*our*)	0.775	0.726	0.892	0.856	0.755	0.681	0.861	0.799	0.671	0.569	0.829	0.714
CharSpan + BPE-Dropout (*our*)	0.775	0.726	0.892	0.856	0.768	0.683	0.877	0.801	0.685	0.582	0.823	0.726

Table 7.12: BLEURT and COMET Scores: HRLs performance for all three language families

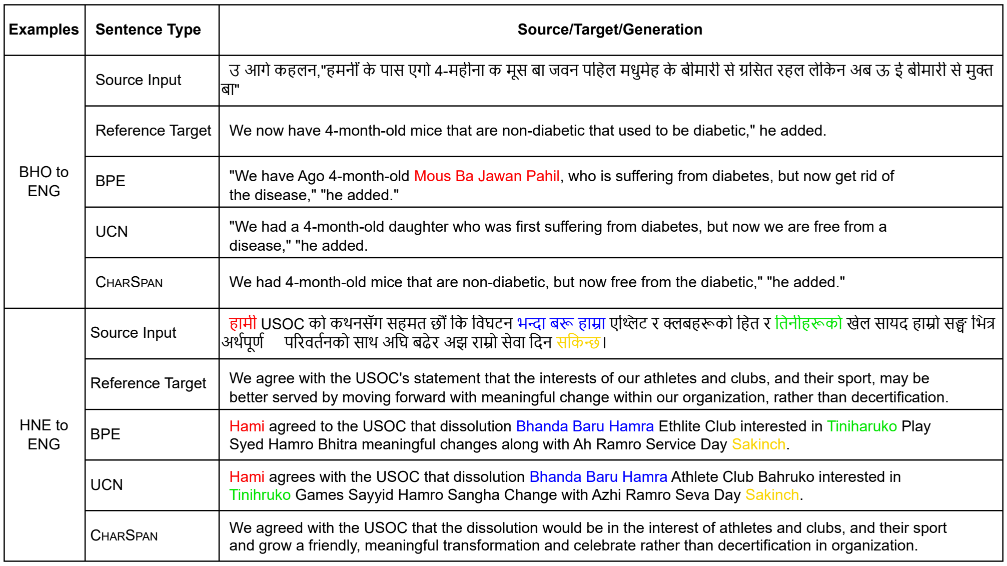
illustrated in Table 7.13.

[Error Analysis - I] Baselines Generations are Transliterated: Fig. 7.7 presents a few sample examples where baseline models have generation errors. Here, we particularly look for transliteration errors. It can observed that many of the source words are directly transliterated in target generation for baseline models; however, the proposed CharSpan model successfully mitigate these error.

[Error Analysis - II] Grammatical Well-Formedness: It is often observed that the generations are grammatically not sound and such features are easily missed by the performance evaluation metrics like ChrF and BLEU. With this error analysis, we aim to investigate the grammatical well-formedness of generations from different

Experimental Setups	BLEU (XX → EN) Gom	Bho	Hne	chrF (XX → EN) Gom	Bho	Hne
char-noise (9%-11% + replacement with only vowels)	4.77	11.21	15.17	28.08	40.36	46.13
char-noise (9%-11%+ replacement with only consonants)	4.79	11.25	15.3	26.95	40.51	46.17
char-noise (9%-11% + replacement with char sound similarity)	4.55	10.7	15.78	27.86	40.45	46.98
char-noise (9%-11% + with number and punctuation)	5.13	12.07	17.66	27.66	41.43	48.68
char-noise (9%-11% + only insertion)	5.04	12.3	17.81	27.50	41.87	48.74
char-noise (9%-11% + only replacement)	5.58	12.8	18.75	28.85	42.43	49.68
char-noise (9%-11%+ only deletion)	4.22	11.92	18.39	28.65	42.02	49.36
char-noise (4%-6% + all three operations + equal probability)	5.44	11.66	18.01	28.62	40.95	48.63
char-noise (14%-16% + all three operations + equal probability)	5.17	11.4	17.01	27.93	40.32	47.61
char-noise (9%-11% + all three operations + equal probability)	5.21	12.62	18.29	28.85	42.53	49.35
char-span noise (9%-11% + 1-3 grams + replacement: N random chars -> span)	3.80	8.80	13.11	25.38	28.22	43.39
char-span noise (9%-11% + 1-3 grams + insertion: 1 random chars -> span)	**5.84**	13.29	20.49	29.29	43.51	51.33
char-span noise (9%-11% + 1-3 grams + insertion: N random chars -> span)	4.81	12.21	17.36	26.98	41.26	47.91
char-span noise (9%-11% + 1-3 grams + all three operations + equal probability)	4.01	10.41	16.33	27.99	36.66	46.13
char-span noise (9%-11% + 1-2 grams + replacement and deletion + equal probability)	5.42	12.08	18.02	29.17	42.21	49.17
char-span noise (9%-11% + 1-4 grams + replacement and deletion + equal probability)	5.79	11.85	18.02	**29.71**	42.41	49.74
char-span noise (9%-11% + 1-5 grams + replacement and deletion + equal probability)	5.56	11.36	17.06	24.13	26.35	29.55
char-span noise (9%-11%+ 1-3 grams + replacement and deletion +unequal probability)	5.48	12.12	18.16	29.01	41.74	49.37
Proposed: char-span noise (9%-11% + 1-3 grams + replacement and deletion + equal probability)	**5.81**	**13.81**	**21.03**	**29.71**	**43.75**	**51.69**

Table 7.13: Ablation Study and Different Experimental Setups. Similar trends were observed for other ELRLs and language families.



Examples	Sentence Type	Source/Target/Generation
BHO to ENG	Source Input	उ आगे कहलन,"हमनीं के पास एगो 4-महीना क मूस बा जवन पहिल मधुमेह के बीमारी से ग्रसित रहल लेकिन अब ऊ इ बीमारी से मुक्त बा"
	Reference Target	We now have 4-month-old mice that are non-diabetic that used to be diabetic," he added.
	BPE	"We have Ago 4-month-old Mous Ba Jawan Pahil, who is suffering from diabetes, but now get rid of the disease," "he added."
	UCN	"We had a 4-month-old daughter who was first suffering from diabetes, but now we are free from a disease," "he added.
	CharSpan	We had 4-month-old mice that are non-diabetic, but now free from the diabetic," "he added."
HNE to ENG	Source Input	हामी USOC को कथनसँग सहमत छौं कि विघटन भन्दा बरू हाम्रा एथ्लेट र क्लबहरूको हित र तिनीहरूको खेल सायद हाम्रो सङ्ग भित्र अर्थपूर्ण परिवर्तनको साथ अघि बढेर अझ राम्रो सेवा दिन सकिन्छ।
	Reference Target	We agree with the USOC's statement that the interests of our athletes and clubs, and their sport, may be better served by moving forward with meaningful change within our organization, rather than decertification.
	BPE	Hami agreed to the USOC that dissolution Bhanda Baru Hamra Ethlite Club interested in Tiniharuko Play Syed Hamro Bhitra meaningful changes along with Ah Ramro Service Day Sakinch.
	UCN	Hami agrees with the USOC that dissolution Bhanda Baru Hamra Athlete Club Bahruko interested in Tinihruko Games Sayyid Hamro Sangha Change with Azhi Ramro Seva Day Sakinch.
	CharSpan	We agreed with the USOC that the dissolution would be in the interest of athletes and clubs, and their sport and grow a friendly, meaningful transformation and celebrate rather than decertification in organization.

Figure 7.7: The generation errors (transliteration) from different baseline models. The proposed CharSpan model successfully mitigates those errors. Colors indicate the corresponding transliteration in a generation.

baseline models. To score the grammatical well-formedness, we use L'AMBRE tool[11]. The results are reported in Table 7.14. For simplicity, we have shown results for only the Indo-Aryan family. The CharSpan shows better Well-Formedness than BPE and Unigram char-noise model across all considered ELRLs.

[11] https://github.com/adithya7/lambre

These error analyses further provide evidence that the performance gains are truly genuine for the CharSpan model.

Models	Indo-Aryan						
	Bho	**Hne**	**San**	**Npi**	**Mai**	**Mag**	**Awa**
BPE	0.9782	0.9813	0.9444	0.9624	0.9647	0.9784	0.9812
UCN	0.9754	0.9616	0.9504	0.9592	0.947	0.9708	0.9753
CharSpan	**0.9856**	**0.9865**	**0.9658**	**0.9735**	**0.9802**	**0.9842**	**0.9836**

Table 7.14: Grammatical Well-Formedness for different models with L'AMBRE

7.3.5 Summary

To summarize, this study presents a simple yet effective novel character-span noise (CSN) argumentation model, CharSpan, to facilitate better cross-lingual transfer from HRLs to closely related ELRLs. The approach generalizes to closely related HRL-ELRL pairs from three typologically diverse language families. The proposed model consistently outperformed all the baselines. To the best of our knowledge, we are the first to apply noise augmentation for the NLG task. The CharSpan model emerged as a state-of-the-art model for ELRLs to English MT task.

7.4 Selective Character Noise Augmentation

7.4.1 Methodology

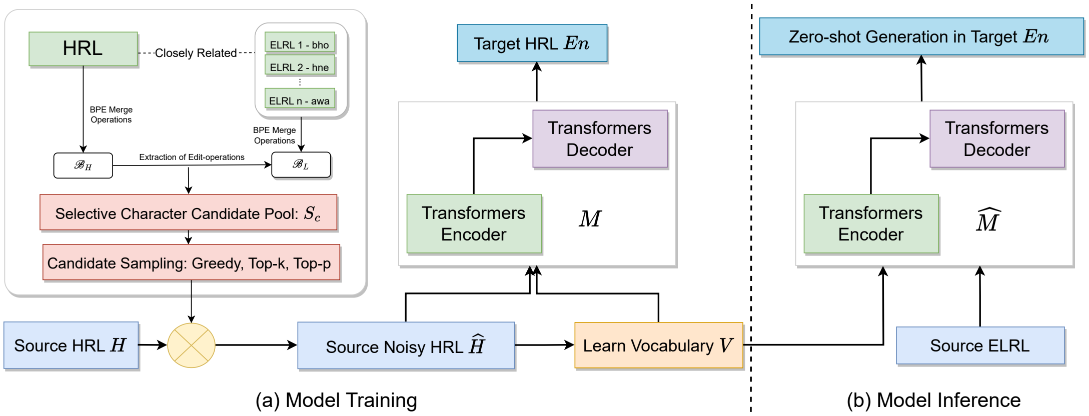


Figure 7.8: Overview of the proposed selective noise augmentation-based model for extremely low-resource MT.

In this section, we present the details of our second proposed model, SelectNoise. In SelectNoise, the noise augmentation candidates are extracted through an unsupervised approach. Specifically, we consider a small monolingual dataset for HRL, denoted as $\mathcal{D}_{\mathcal{H}}$, as well as related (lexically similar) LRLs, denoted as $\mathcal{D}_{\mathcal{L}}$. During the process of building the Byte Pair Encoding (BPE) vocabulary, we extract the BPE operations separately for the HRL ($\mathcal{B}_{\mathcal{H}}$) and ELRLs ($\mathcal{B}_{\mathcal{L}}$). Generally, $\mathcal{B}_{\mathcal{L}}$ comprises small monolingual datasets from multiple extremely ELRLs. Next, we design an algorithm $\mathcal{A}$ to extract selective candidates $\mathcal{S}_{\mathcal{C}}$ from $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{B}_{\mathcal{L}}$, inspired by an edit-operation approach. In other words, we obtain $\mathcal{S}_{\mathcal{C}}$ as a result of $\mathcal{A}$ ($\mathcal{B}_{\mathcal{H}}$, $\mathcal{B}_{\mathcal{L}}$). These selective candidates $\mathcal{S}_{\mathcal{C}}$ are augmented into the source sentences of HRL corpus ($\mathcal{H}$) from the large parallel dataset $\mathcal{P}_{\mathcal{H}} = \{(h, e) | lang(h) = \mathcal{H}, lang(e) = En\}$ using a noising mechanism η, resulting in a noise augmented dataset $\hat{\mathcal{P}_{\mathcal{H}}} = \{(\hat{h}, e) | lang(\hat{h}) = \hat{\mathcal{H}}, lang(e) = En\}$, where $\hat{\mathcal{H}} = \eta(\mathcal{H})$. We train the stranded encoder-decoder transformer model ($\mathcal{M}$; [VSP+17]) from scratch with $\hat{\mathcal{H}}$ and obtained and trained model $\hat{\mathcal{M}}$. Finally, zero-shot evaluation is done for ELRLs with $\hat{\mathcal{M}}$. In the next subsections, we will deep dive into each component of the proposed model, which includes unsupervised selective candidate extraction based on edit operations and BPE merge operations, candidate noise augmentation approach based on a sampling strategy, model training, and zero-shot evaluation. The overview of the proposed approach is depicted in Figure 7.8.
In this work, we investigate two hypotheses: (a) *the selective noise augmentation strategy is expected to outperform random noise augmentation*, and (b) *the performance of the unsupervised noise augmentation model should be comparable to that of the supervised noise augmentation model that utilizes parallel data.*

Unsupervised Noise Augmentation

The formal procedure for unsupervised noise augmentation is presented in Algorithm 3. In the next subsections, we will dive deep into each stage of the proposed model in detail:

Selective Candidate Extraction: The first stage in the proposed approach involves extracting candidate characters that will subsequently be utilized for noise augmentation. Given $\mathcal{D}_{\mathcal{H}}$ and $\mathcal{D}_{\mathcal{L}}$, we extract all BPE merge operations $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{B}_{\mathcal{L}}$, respectively. Each merge operation consists of tuples $\langle (p, q) \rangle \in \mathcal{B}_{\mathcal{H}}$ and $\langle (r, s) \rangle \in \mathcal{B}_{\mathcal{H}}$. We pair each merge tuple of $\mathcal{B}_{\mathcal{H}}$ with each tuple of $\mathcal{B}_{\mathcal{L}}$ (i.e., cartesian setup). If $\mathcal{B}_{\mathcal{H}}$ and $\mathcal{B}_{\mathcal{L}}$ have n and m merge operations, respectively, we obtain a total of $t = m \cdot n$ pairs. We consider only those pairs where either p and r or q and s are the same

```
Algorithm 3 SelectNoise: Unsupervised Noise Augmentation
Require: [Inputs] HRL monolingual data $\mathcal{D}_{\mathcal{H}}$; closely related ELRLs monolingual data
    $\mathcal{D}_{\mathcal{L}}$; number of merge operations $\mathcal{M}_{\mathcal{O}}$; HRL parallel data $\mathcal{P}_{\mathcal{H}}(\mathcal{H}, En)$; Noise augmentation percentage range [$p_1$% - $p_2$%]; candidate sampling strategy $\mathcal{S}_{\mathcal{M}}$; ExtractSelectiveCands ($\mathcal{A}$)
Ensure: [Output] Noisy source HRL $\hat{\mathcal{H}}$
$\mathcal{S}_c$ = ExtractSelectiveCands($\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathcal{L}}, \mathcal{M}_{\mathcal{O}}$)
for sentence $s$ in $\mathcal{H}$ do
    $idxs \leftarrow$ randomly select [$p_1$% - $p_2$%] indices of $s$
    for $idx$ in $idxs$ do
        $ops \leftarrow$ randomly sample operation {insert, delete, substitute}
        if $ops$ equals delete then
            Remove character at index $idx$
        end if
        if $ops$ equals Insert or $ops$ equals substitute then
            $c$ = sample candidate char, i.e., $\mathcal{S}_{\mathcal{M}}(\mathcal{S}_c$, $ops)$
            Perform operation $ops$ at index $idx$ with $c$
        end if
    end for
end for
procedure ExtractSelectiveCands($\mathcal{D}_{\mathcal{H}}, \mathcal{D}_{\mathcal{L}}, \mathcal{M}_{\mathcal{O}}$)
    Initialize candidate pool $\mathcal{S}_c \leftarrow \emptyset$ to store candidates
    Compute merge operations $\mathcal{B}_{\mathcal{H}}$ = BPE($\mathcal{D}_{\mathcal{H}}$, $\mathcal{M}_{\mathcal{O}}$)
    Compute merge operations $\mathcal{B}_{\mathcal{L}}$ = BPE($\mathcal{D}_{\mathcal{L}}$, $\mathcal{M}_{\mathcal{O}}$)
    for $n$ in $\mathcal{B}_{\mathcal{H}}$ do
        for $m$ in $\mathcal{B}_{\mathcal{L}}$ do
            if $n$ equals $m$ or ($p$ not equals $r$ and $q$ not equals $s$) then
                                        ▷ where $n$ = tuple $\langle(p, q)\rangle$, $m$ = tuple $\langle(r, s)\rangle$
                No operation is performed with $n$ and $m$
            else if $p$ equals $r$ then
                Compute edit-operations($q$, $s$) & update $\mathcal{S}_c$
            else if $q$ equals $s$ then
                Compute edit-operations($p$, $r$) & update $\mathcal{S}_c$
            end if
        end for
    end for
    return $\mathcal{S}_c$
end procedure
```

while discarding the rest. For the considered tuples $\langle(p, q), (r, s)\rangle$, we calculate the character-level edit-distance operations between non-similar elements of the tuple. For instance, if p and r are the same, the edit operations are obtained using q and s elements. These operations are collected in the candidate pool $\mathcal{S}_c$, which includes *insertions*, *deletions*, and *substitutions*, and are referred to as the *selective candidates*.

As discussed, the extracted selective candidates are stored in the candidate pool $\mathcal{S}c$, a dictionary data structure encompassing HRL and ELRL characters. The $\mathcal{S}c$ consists of HRL characters, ELRL characters, edit operations, and their respective frequencies. An element of $\mathcal{S}c$ has following template: $c_i : \{I : f_{ins}, D : f_{del}, S : \{c'_1 : f_1, c'_2 : f_2, c'_k : f_k\}\}$. The operations are: *insertion* (I), *deletion* (D) and *substitution* (S). The character c_i represents the i^{th} element of $\mathcal{S}_c$, which is an HRL character, $c'_1 \dots c'_k$ denote the corresponding substituting candidates from ELRLs and f is the associated frequencies. A few examples of selective candidate extraction are illustrated in Fig. 7.9. Sample candidate pool ($\mathcal{S}_c$) is shown in Fig. 7.10.

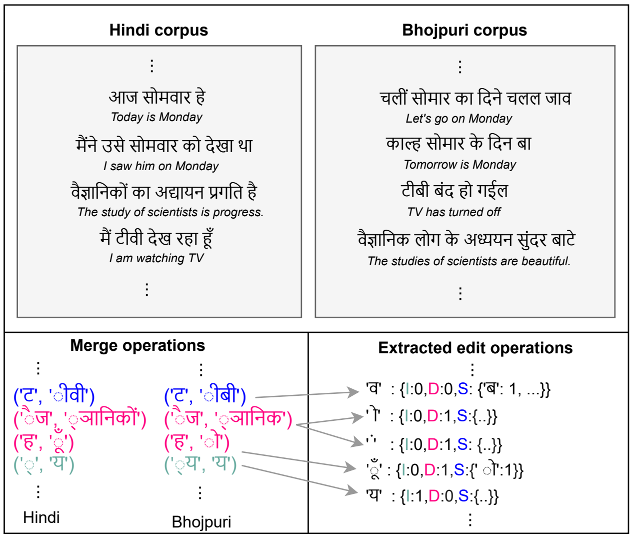
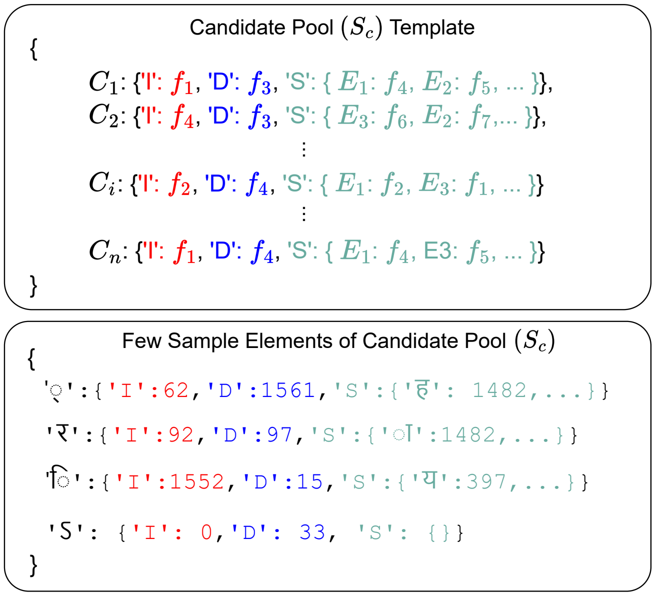


Figure 7.9: Illustration of selective candidates extraction for noise augmentation that utilizes BPE merge and edit operations. Here I, D, and S indicate insertion, deletion, and substitution respectively. Frequencies are associated with operations. 0 indicates the corresponding edit operation was not extracted.

Intuitively, with this candidate pool $\mathcal{S}_c$, we have learned transformative entities that can be used to resemble an HRL to lexically similar ELRLs, which results in bridging the lexical gap between HRL and LRLs. Training with such modified HRL data enhances the effectiveness of cross-lingual transfer signals for ELRLs. As candidates are extracted by considering the vocabulary word-formation s'trategy from BPE and edit operations, they indirectly consider the linguistic cues/information.

Noise augmentation to HRL: In the second stage, we sample selective candidates from $\mathcal{S}_c$ and augment into the source sentences of HRL corpus ($\mathcal{H}$) from the parallel dataset $\mathcal{P}_{\mathcal{H}} = \{(h, e) | lang(h) = \mathcal{H}, lang(e) = En\}$ using a noise function η,



Candidate Pool (S_c) Template

{

C_1: {'I': f_1, 'D': f_3, 'S': { E_1: f_4, E_2: f_5, ... }},

C_2: {'I': f_4, 'D': f_3, 'S': { E_3: f_6, E_2: f_7,... }},

$\vdots$

C_i: {'I': f_2, 'D': f_4, 'S': { E_1: f_2, E_3: f_1, ... }}

$\vdots$

C_n: {'I': f_1, 'D': f_4, 'S': { E_1: f_4, E3: f_5, ... }}

}

Few Sample Elements of Candidate Pool (S_c)

```
{
 '◌्':{'I':62,'D':1561,'S':{'ह': 1482,...}}
 'र':{'I':92,'D':97,'S':{'◌ा':1482,...}}
 '◌ि':{'I':1552,'D':15,'S':{'य':397,...}}
 'ऽ': {'I': 0,'D': 33, 'S': {}}
}
```

Figure 7.10: *Top* is a template for the character candidate pool $\mathbf{S}_c$. The operations are: *insertion* (I), *deletion* (D) and *substitution* (S). The character c_i represents the i^{th} element of $\mathcal{S}_c$, which is an HRL character, $c'_1 \ldots c'_k$ denote the corresponding substituting candidates from ELRLs and f is the associated frequencies. The *Bottom* shows a few sample elements of the $\mathbf{S}_c$.

resulting in a noise augmented (augmented) parallel dataset $\hat{\mathcal{P}_{\mathcal{H}}} = \{(\hat{h}, e) | lang(\hat{h}) = \hat{\mathcal{H}}, lang(e) = En\}$, where $\hat{\mathcal{H}} = \eta(\mathcal{H})$. Details of the noise function and candidate sampling strategy are presented below:

- **Noise Function:** The noise augmentation function (η) is designed as follows: Initially, we randomly select 5%-10%[12] of character indices from a sentence $s \in \mathcal{H}$. Subsequently, we uniformly choose between *insertion*, *deletion*, or *substitution* operations with equal probability. If the selected operation is insertion or substitution, we sample a candidate character from $\mathcal{S}_c$ to perform the noise augmentation operation. For deletion, the charter is simply deleted. These steps are repeated for all sentences in $\mathcal{H}$ to obtain the final $\hat{\mathcal{H}}$.

- **Candidate Character Sampling:** While noise augmentation for deletion operation, we simply delete the character. For insertion and substitution, we sample the candidate character for augmentation from $\mathcal{S}_c$ using the *greedy*, *top-p* (nucleus), and *top-k* sampling techniques inspired by decoding algorithms com-

[12] after conducting several ablation experiments, this range provides the best performance

monly employed in NLG [HBFC19]. Before applying these sampling techniques, the frequencies of the candidate characters are transformed into probability scores using the softmax operation. Intuitively, with the sampling technique, we aim to explore not only frequent candidate characters but also diverse candidates.

The performance of any learning model depends on the quality of the training data. The presence of noise hampers the learning, and the outputs of the learned model exhibit the different nuances of the noise present in the data. In our specific case: (i) We train a model using data that contains noise, resulting in the model's increased robustness to minor lexical variations in different languages, particularly those related to ELRLs. (ii) The noise is added for a small portion of characters (5-10%), making the HRLs training data closely resemble how sentences appear in ELRLs. As a result, the model is able to do a robust cross-lingual transfer to the ELRL in a zero-shot setting. In another perspective, the augmentation of noise acts as a regularizer [AS22], contributing to an overall enhancement in the model's performance.

Supervised Noise augmentation

We have also investigated in a supervised setting akin to the proposed SelectNoise approach. The key distinction lies in how the candidate pool $\mathcal{S}sc$ is derived from a limited parallel dataset between HRL and ELRLs. For each HRL and ELRL pair, we extract a candidate set using edit operations and subsequently combine all the candidate sets in $\mathcal{S}sc$. The rest of the modeling steps are similar to the SelectNoise. We hypothesize that the unsupervised method should exhibit competitive performance compared to the supervised approach. In the supervised candidate extraction, we assume the availability of a limited amount of parallel data of approximately 1000 examples. A formal algorithm outlining in the Algorithm 4.

Model Training and Zero-shot Evaluation

The stranded encoder-decoder transformers model ($\mathcal{M}$) is trained from scratch using the noisy high-resource parallel dataset $\hat{\mathcal{P}_{\mathcal{H}}}$ and $\mathcal{V}$ to obtain a trained model $\hat{\mathcal{M}}$. Where $\mathcal{V}$ is learned BPE vocabulary with $\hat{\mathcal{P}_{\mathcal{H}}}$. Subsequently, we use $\hat{\mathcal{M}}$ to perform zero-shot generation for ELRLs. We have not used any parallel training data for ELRLs and directly employ $\hat{\mathcal{M}}$ for inference, making this modeling setup zero-shot. The trained model transfers knowledge across languages, enabling coherent and meaningful translation for ELRLs.

Algorithm 4 Supervised Noise Augmentation

```
Require: [Inputs] joint parallel data for all considered HRL-ELRL pairs $\mathcal{P}_{\mathcal{S}}(\mathcal{S}, \mathcal{E}_L)$; HRL
    parallel data $\mathcal{P}_{\mathcal{H}}(\mathcal{H}, En)$; noise augmentation percentage range [$p_1$% - $p_2$%]; candidate
    sampling strategy $\mathcal{S}_{\mathcal{M}}$
Ensure: [Output] Noisy source HRL $\hat{\mathcal{H}}$
$\mathcal{S}_{sc}$ = SupExtractSelectiveCands($\mathcal{P}_{\mathcal{S}}$)
for sentence $s$ in $\mathcal{H}$ do
    $idxs \leftarrow$ randomly select [$p_1$% - $p_2$%] indices of $s$
    for $idx$ in $idxs$ do
        $ops \leftarrow$ randomly sample operation {insert, delete substitute}
        if $ops$ equals delete then
            Remove character at index $idx$
        end if
        if $ops$ equals Insert or $ops$ equals substitute then
            $c$ = sample candidate char, i.e., $\mathcal{S}_{\mathcal{M}}(\mathcal{S}_{sc}, ops)$
            Perform operation $ops$ at index $idx$ with $c$
        end if
    end for
end for
procedure SupExtractSelectiveCands($\mathcal{P}_{\mathcal{S}}$)
    Initialize candidate pool $\mathcal{S}_{sc} \leftarrow \emptyset$ to store candidates
    for each $\langle(s, e)\rangle$ in $\mathcal{P}_{\mathcal{S}}$ do
        Compute edit-operations($s$, $e$) & update $\mathcal{S}_{sc}$
    end for
    return $\mathcal{S}_{sc}$
end procedure
```

7.4.2 Experimental Setup

We designed our experimental setup to address the following set of questions: (1) Does noise augmentation improve performance for NLG tasks, i.e., MT in our case? (2) Does selective noise augmentation with the proposed SelectNoise model outperform the random noise augmentation model [AS22]? (3) Does the model's performance persist across different language families? and (4) Does the unsupervised SelectNoise model demonstrate competitive performance with supervised approach? Based on these research questions, we have designed our experimental setup. As the CharSpan is closely related work, for clarity, some experimental setup is repeated, and readers are suggested to skip to those sections.

Datasets

The primary constraint of the proposed approach is to select closely related HRLs and ELRLs. With this criterion in mind, we have chosen two language families: *Indo-*

ELRL/HRL-Pair	ISO-3 Code	Language Family	Train	Valid	Test	HRL	Source
Bhojpuri	Bho	Indo-Aryan	-	997	1012	hi	FLORES-200
Magahi	Mag	Indo-Aryan	-	997	1012	hi	FLORES-200
Maithili	Mai	Indo-Aryan	-	997	1012	hi	FLORES-200
Nepali	Npi	Indo-Aryan	-	997	1012	hi	FLORES-200
Awadhi	Awa	Indo-Aryan	-	997	1012	hi	FLORES-200
Sanskrit	San	Indo-Aryan	-	997	1012	hi	FLORES-200
Kashmiris	Kas	Indo-Aryan	-	997	1012	hi	FLORES-200
Chhattisgarhi	Hne	Indo-Aryan	-	997	1012	hi	FLORES-200
Asturian	Ast	Romance	-	997	1012	es	FLORES-200
Catalan	Cat	Romance	-	997	1012	es	FLORES-200
Galician	Glg	Romance	-	997	1012	es	FLORES-200
Occitan	Oci	Romance	-	997	1012	es	FLORES-200
Hindi-English	hi-en	Indo-Aryan	10.1M	997	1012	-	[RDB+22]
Spanish-English	es-en	Romance	6.6M	997	1012	-	[Rap21]

Table 7.15: Statistics of the language and data used in SelectNoise model

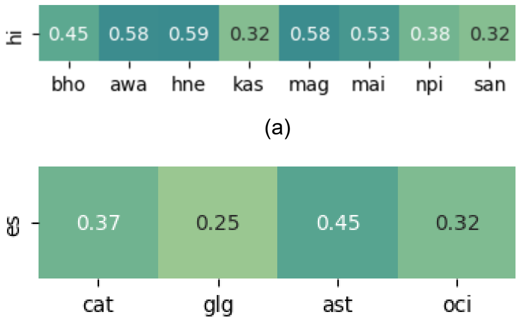
Aryan and *Romance*. Within the Indo-Aryan family, we have selected Hindi (Hi) as the HRL and 8 ELRLs were Awadhi (Awa), Bhojpuri (Bho), Chhattisgarhi (Hne), Kashmiri (Kas), Magahi (Mag), Maithili (Mai), Nepali (Npi), and Sanskrit (San), based on their lexical similarity. For the Romance family, Spanish (Es) served as the HRL, and the 4 ELRLs were Asturian (Ast), Catalan (Cat), Galician (Glg), and Occitan (Oci). We conducted separate experiments for each language family, training the model with the HRL to English MT task and evaluating it in a zero-shot setting with corresponding ELRLs.

In total, we have 3 HRLs (English, Hindi, and Spanish) and 12 ELRLs. All the test datasets are sourced from FLORES-200 [CjCÇ+22], while the hi-en dataset is obtained from AI4Bharat [RDB+22], and the es-en dataset is from Rapp et al. [Rap21]. The development set of FLORES-200 was utilized as a parallel dataset for supervised noise augmentation. A small amount of monolingual data was used for SelectNoise and other baseline methods. Here, we used 1000 examples for each ELRL. Detailed dataset statistics and data sources are presented in Table 7.15. Fig. 7.11, we provide an overview of the lexical similarity between HRLs and ELRLs.

Baselines

We compare the SelectNoise model with several strong baselines, including a traditional data augmentation model, lexical similarity-based models, and a model based on random noise augmentation. Details of each baseline are presented below:

- **Vanilla NMT:** [SHB16b] A standard transformer-based NMT model with BPE algorithm .



(a)

(b)

Figure 7.11: Lexical similarity heatmap between HRL and its related ELRLs. Fig. (a) depicts a similarity score for the Indo-Aryan family where HRL is Hindi. Fig. (b) depicts a similarity score for the Romance family where HRL is Spanish. *Note: Darker color denotes high lexical similarity.*

- **Word-drop [SHB16a]:** In this baseline, the embeddings of randomly selected 10% words in the source sentences of HRL to 0. The rest of the steps are similar to the SelectNoise model.

- **BPE-drop:** This approach is similar to the word-drop baseline but uses BPE tokens instead of words.

- **SwitchOut [WPDN18]:** This baseline employs a data augmentation technique where random words in both the source and target sentences are replaced with randomly selected words from their respective vocabularies. We have utilized the officially released implementation with a 10% word replacement rate.

- **OBPE [PTS22]:** The approach modifies the BPE algorithm to encourage more shared tokens from HRLs and LRLs in the vocabulary. This model required a monolingual dataset for LRLs. We use a small monolingual dataset, based on availability, for the ELRLs. Earlier work applied OBPE for NLU tasks only - we are the first to investigate it for MT.

- **BPE Dropout [PEV20]:** It is based on the BPE algorithm to learn the vocabulary and generates non-deterministic segmentations for input text *on-the-fly* during training. We use a dropout value of 0.1.

- **Random Char Noise [AS22]:** This baseline methodology is similar to the proposed SelectNoise approach; but, noise augmentations are done randomly.

Evaluation Metrics

All the model performances are compared using both automated and human evaluation metrics. In line with recent research on MT for LRLs, we employ two types of automated evaluation metrics [CjCÇ+22, SBF+22]. Specifically, lexical match-based metrics: BLEU [PRWZ02d] and chrF [Pop15] and learning-based metrics: BLEURT [SDP20] and COMET [PCP+21].

We further conducted the human evaluation to ensure the reliability of the performance gain. Three languages from the Indo-Aryan family (Bhojpuri, Nepali, and Sanskrit) were selected based on their high, moderate, and low lexical similarity with the HRL (Hindi). To manage the annotators' workload effectively, we limited our evaluation to three models: Vanilla NMT, BPE Dropout, and SelectNoise. For each language, the human evaluation set consisted of 24 examples, and translations were obtained from above mentioned three models. Two annotators were employed for each language to ensure the inter-annotator agreement, and two ratings were obtained for each example from these annotators. All annotators held at least a master's degree, were native speakers of the respective language and demonstrated proficiency in English. We use *Crosslingual Semantic Text Similarity (XSTS)* metric [ACDGA12], which is widely adopted in the MT research for human evaluation. The XSTS metric employs a 1-5 evaluation scale, where 1 represents a very bad translation and 5 represents a very good translation.

Implementation Details

Our vanilla NMT model is based on standard transformer architecture consisting of 6 encoder and decoder layers. We trained our model for a maximum epoch of 15. We use Adam [KB15] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We set a learning rate of 0.0005. We use a dropout of 0.2. We performed data normalization and preprocessing using IndicNLP library[13]. We perform our experiments using fairseq[14] library. For evaluation we use the lexical match-based BLEU metric[15] [PRWZ02d], chrF[16] [Pop15] metric, semantic-based BLEURT[17] [SDP20], and COMET[18] [PCP+21] metrics.

[13] https://github.com/anoopkunchukuttan/indic_nlp_library

[14] https://github.com/pytorch/fairseq

[15] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

[16] nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1

[17] Reported using BLEURT20 checkpoint

[18] Reported using wmt22-comet-da model

7.4.3 Results and Discussions

Models	Indo-Aryan								Romance				Average
	Bho	Hne	San	Mai	Mag	Awa	Npi	Kas	Cat	Glg	Ast	Oci	
Vanilla NMT	40.3	46.8	22.3	40.0	49.3	47.6	29.6	21.3	33.0	41.0	40.7	33.0	37.08
Word-drop	39.5	47.2	21.8	40.6	49.0	47.6	28.6	20.6	37.6	43.6	43.4	36.0	37.96
BPE-drop	39.1	46.8	22.6	40.4	48.7	46.7	29.2	21.1	33.8	41.7	41.5	33.0	37.05
SwitchOut	36.1	43.2	20.1	38.2	45.6	42.7	28.3	18.8	29.0	34.9	34.9	29.1	33.41
OBPE	41.3	47.5	23.4	41.8	50.4	49.7	30.5	21.1	34.1	41.2	41.3	33.8	38.00
BPE-Dropout	39.8	47.4	22.5	39.9	49.6	47.7	29.3	21.2	33.2	40.8	41.4	33.0	37.15
Random Char Noise	40.9	48.4	23.8	40.8	50.0	47.5	31.2	21.9	40.9	46.1	46.4	38.2	39.68
SelectNoise Model													
SelectNoise + Greedy	42.1	**51.0**	25.2	**43.4**	**51.7**	**49.9**	33.4	**23.7**	**42.0**	**47.1**	47.4	38.5	**41.28**
SelectNoise + Top-k	**42.4**	49.9	**26.0**	43.0	51.0	48.8	33.4	23.3	41.5	47.1	**47.8**	38.5	41.06
SelectNoise + Top-p	42.0	49.6	24.1	42.4	50.6	48.8	**33.6**	23.3	41.6	47.1	47.5	**38.8**	40.78
Supervised Noise augmentation Model													
Selective noise + Greedy	41.4	49.1	25.4	42.2	50.1	48.7	32.9	22.2	41.6	47.2	47.7	38.7	40.60
Selective noise + Top-k	41.7	49.3	26.3	43.3	50.8	48.7	34.2	23.6	41.9	46.8	47.5	38.7	41.10
Selective noise + Top-p	41.4	49.9	27.3	43.3	51.6	48.9	33.9	23.4	41.6	47.7	48.2	39.0	41.35

Table 7.16: Zero-shot chrF (↑) scores results for ELRLs → English

Models	Indo-Aryan								Romance				Average
	Bho	Hne	San	Mai	Mag	Awa	Npi	Kas	Cat	Glg	Ast	Oci	
Vanilla NMT	11.1	17.2	2.7	10.1	18.5	18.3	5.1	2.6	5.3	10.1	12.3	5.2	9.86
Word-drop	8.7	13.7	1.9	7.7	15.2	16.1	3.0	1.6	6.9	10.7	13.3	6.5	8.76
BPE-drop	10.8	16.1	2.7	10.0	17.2	17.8	4.0	2.1	5.1	9.1	11.2	4.7	9.23
SwitchOut	4.3	7.7	1.4	4.9	8.4	7.9	2.9	1.2	3.5	6.3	8.2	3.8	5.04
OBPE	11.1	16.6	2.9	10.4	18.7	19.7	4.8	1.9	6.2	10.7	12.9	6.1	10.16
BPE-Dropout	11.6	17.5	3.1	10.1	19.3	18.3	5.4	2.5	5.4	10.1	13.0	5.4	10.14
Random Char Noise	**12.8**	18.8	3.1	10.2	19.4	18.6	6.3	2.9	**10.9**	14.3	17.2	8.7	11.93
SelectNoise Model													
SelectNoise + Greedy	12.5	**20.1**	3.7	11.9	**21.2**	**20.2**	7.1	3.0	10.8	**15.0**	17.4	**9.0**	**12.66**
SelectNoise + Top-k	12.3	19.7	**3.8**	**12.0**	20.2	19.5	**7.2**	2.8	10.5	**15.0**	**17.5**	8.8	12.44
SelectNoise + Top-p	12.7	19.5	3.8	11.9	20.3	19.6	6.7	**3.2**	10.7	14.8	17.1	8.9	12.43
Supervised Noise augmentation Model													
Selective noise + Greedy	13.1	19.5	4.0	11.8	19.6	19.3	6.8	2.4	10.5	15.0	17.9	8.9	12.4
Selective noise + Top-k	12.7	19.1	3.9	12.2	20.1	19.3	7.0	2.9	10.8	15.0	17.4	8.9	12.44
Selective noise + Top-p	12.7	20.0	4.1	12.6	21.2	19.7	7.0	2.7	10.5	15.4	18.1	9.1	12.76

Table 7.17: Zero-shot BLEU (↑) scores results for ELRLs → English

Models	Indo-Aryan								Romance				Average
	Bho	Hne	San	Mai	Mag	Awa	Npi	Kas	Cat	Glg	Ast	Oci	
Vanilla NMT	0.500	0.531	0.368	0.500	0.559	0.576	0.435	0.377	0.295	0.390	0.406	0.232	0.431
Word-drop	0.497	0.533	0.357	0.498	0.551	0.563	0.417	0.353	0.361	0.440	0.454	0.312	0.445
BPE-drop	0.506	0.537	0.367	0.509	0.557	0.572	0.422	0.363	0.316	0.415	0.432	0.283	0.440
SwitchOut	0.411	0.446	0.318	0.415	0.467	0.466	0.38	0.335	0.278	0.337	0.347	0.262	0.372
OBPE	0.502	0.525	0.371	0.502	0.561	0.583	0.436	0.381	0.306	0.404	0.416	0.266	0.438
BPE-Dropout	0.501	0.526	0.371	0.497	0.558	0.574	0.439	0.393	0.300	0.389	0.410	0.231	0.432
Random Char Noise	0.521	0.547	0.371	0.501	0.569	0.584	0.441	0.380	0.391	0.487	0.491	0.319	0.467
SelectNoise Model													
SelectNoise + Greedy	0.525	0.563	**0.386**	**0.511**	**0.578**	**0.606**	**0.458**	**0.394**	0.392	0.499	0.511	0.319	0.478
SelectNoise + Top-k	0.524	0.558	**0.386**	0.507	0.576	0.599	0.454	0.388	**0.400**	0.497	**0.516**	**0.321**	0.477
SelectNoise + Top-p	**0.527**	**0.599**	0.372	0.505	0.573	0.599	0.457	0.391	0.399	**0.501**	0.509	**0.321**	**0.479**
Supervised Noise augmentation Model													
Selective noise + Greedy	0.527	0.560	0.389	0.507	0.572	0.600	0.451	0.381	0.392	0.499	0.511	0.319	0.476
Selective noise + Top-k	0.526	0.549	0.401	0.509	0.573	0.463	0.463	0.390	0.400	0.494	0.506	0.326	0.467
Selective noise + Top-p	0.524	0.558	0.400	0.510	0.584	0.455	0.455	0.386	0.391	0.501	0.512	0.321	0.466

Table 7.18: Zero-shot BLEURT (↑) scores results for ELRLs → English

In this section, we will discuss results, observations and findings. The zero-shot automated evaluation scores are presented in Tables 7.16, 7.17, 7.18 and 7.19. The results are reported with greedy, top k (k = 50), and top-p (p = 0.25) sampling

Models	**Indo-Aryan**								**Romance**				**Average**
	Bho	**Hne**	**San**	**Mai**	**Mag**	**Awa**	**Npi**	**Kas**	**Cat**	**Glg**	**Ast**	**Oci**	
Vanilla NMT	0.642	0.676	0.471	0.621	0.711	0.736	0.542	0.387	0.499	0.534	0.497	0.408	0.560
Word-drop	0.659	0.702	0.494	**0.650**	0.725	0.747	0.564	0.409	0.484	0.551	0.538	0.421	0.579
BPE-drop	0.653	0.687	0.497	0.645	0.711	0.732	0.554	0.400	0.438	0.515	0.505	0.389	0.560
SwitchOut	0.565	0.605	0.462	0.564	0.626	0.632	0.533	0.394	0.405	0.461	0.445	0.362	0.504
OBPE	0.664	0.676	0.452	0.630	0.707	0.740	0.544	0.392	0.456	0.524	0.501	0.400	0.557
BPE-Dropout	0.644	0.672	0.471	0.616	0.710	0.733	0.537	0.381	0.503	0.534	0.500	0.411	0.559
Random Char Noise	0.673	0.700	0.492	0.641	0.725	0.746	0.559	0.401	0.522	0.610	0.584	0.441	0.591
SelectNoise Model													
SelectNoise + Greedy	0.672	**0.714**	0.493	0.647	**0.735**	**0.765**	0.575	0.412	0.523	0.620	0.598	0.434	0.599
SelectNoise + Top-k	**0.678**	0.708	**0.504**	0.649	0.730	0.758	0.585	**0.419**	0.524	0.621	**0.603**	0.438	**0.601**
SelectNoise + Top-p	0.677	0.559	0.502	0.643	0.730	0.758	**0.586**	0.411	**0.526**	**0.625**	0.600	**0.442**	0.588
Supervised Noise augmentation Model													
Selective noise + Greedy	0.681	0.711	0.505	0.649	0.728	0.761	0.582	0.411	0.522	0.618	0.603	0.441	0.601
Selective noise + Top-k	0.677	0.700	0.506	0.655	0.703	0.757	0.581	0.414	0.522	0.623	0.605	0.439	0.598
Selective noise + Top-p	0.680	0.708	0.511	0.655	0.738	0.756	0.589	0.414	0.522	0.623	0.605	0.439	0.603

Table 7.19: Zero-shot COMET (↑) scores results for ELRLs → English

Models	**ELRLs**		
	Bho	**San**	**Npi**
Annotator set-1			
Vanilla NMT	3.54	2.42	2.21
BPE Dropout	3.29	2.37	1.83
SelectNoise	**4.17**	**2.83**	**2.50**
Annotator set-2			
Vanilla NMT	3.42	1.96	2.17
BPE Dropout	2.79	1.83	1.96
SelectNoise	**3.54**	**2.17**	**2.21**

Table 7.20: Human evaluation scores with XSTS metrics

strategies. Table 7.20 presents the human evaluation results.

SelectNoise vs. Baselines: The proposed and other models that incorporate lexical similarity have demonstrated superior performance compared to the Vanilla NMT model. While general data augmentation techniques like Word-drop and SwitchOut exhibit performance similar to the Vanilla NMT model, they perform poorly when compared to OBPE and BPE-Dropout models. These results indicate the importance of considering monolingual data from ELRLs in the modeling However, random noise augmentation and the SelectNoise approach outperform the OBPE and BPE-Dropout models, indicating the effectiveness of noise augmentation-based modeling techniques. In conclusion, the careful selection of noise candidates, as done in the SelectNoise approach, has outperformed the random noise model (second best) and emerged as the state-of-the-art model.

Selective vs. Random Noise Augmentation: Unsupervised selective noise augmentation approaches exhibit a larger performance gain compared to the random noise augmentation model. This observation emphasizes the importance of a systematic selective candidate extraction and noise augmentation process.

Lexical vs. Learned Evaluation Metrics: We observe a strong correlation between lexical match-based metrics, such as BLEU and chrF scores. Further, semantic-based metrics like BLEURT and COMET exhibit similar trends to lexical match metrics, indicating a high level of correlation. This emphasizes the reliability of evaluation scores.

Automated vs. Human Evaluation: The proposed SelectNoise model outperforms both baselines in human evaluation across all three languages. The model demonstrates acceptable zero-shot performance for ELRLs, with a strong correlation with automated evaluation scores.

Performance across Language Families: Unsupervised selective noise augmentation consistently outperforms all the baselines across ELRLs, with few exceptions. The model exhibits similar performance trends across both language families.

Unsupervised vs. Supervised Noise augmentation: The unsupervised SelectNoise model performs comparably to the supervised model, with slight variations depending on the language and family. The performance gap between the two models is minimal, indicating their equal strength.

Performance vs. Sampling Strategies: The performance with different sampling techniques is compared, and it is observed that the greedy approach for SelectNoise performs better for the majority of languages. This finding indicates the existence of one-to-one lexical mapping across HRL and ELRLs. However, other sampling approaches are also effective for a subset of ELRLs.

Overall Performance: As we can observe from the average automated evaluation scores, the proposed SelectNoise model outperforms all the baselines by a significant margin. It also exhibits comparable performance to the supervised model, and this performance persists across different language families. These findings satisfy our hypothesis, leading us to conclude that the proposed SelectNoise model is a state-of-the-art model for English-to-ELRLs MT systems.

Further Analyses

In this section, we perform a detailed analysis with SelectNoise to understand factors contributing to performance gain and analyze robustness.

Performance Trend with Top-k and Top-p: In Figure 7.12, the performance trend of the proposed model with varying values of k and p for top-p and top-k sampling is depicted. The candidate pool consists of a maximum of 61 characters (a range for k-value selection). The model performs best with a k-value of 50 and a p-value of 0.25, offering valuable insights for optimizing its performance through parameter selection.

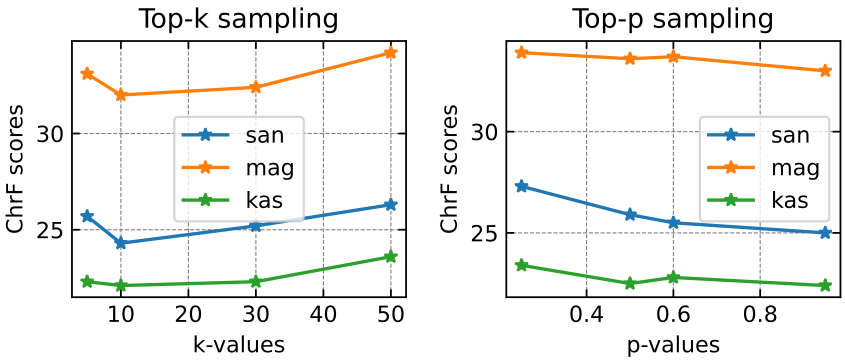


Figure 7.12: Performance trends of the proposed model with various k and p values from top-k and top-p sampling, respectively.

Impact of Monolingual data size: The proposed SelectNoise model relies on the small monolingual dataset of ELRLs. We investigate the impact of a large monolingual dataset on the model's performance for ELRLs. Table 7.21 demonstrates that a larger dataset leads to a performance boost, suggesting the extraction of more meaningful noise augmentation candidates.

Language similarity Vs. Performance: Figure 7.13 illustrates the comparative trend of lexical similarity score between ELRLs and HRLs and performance (ChrF score). It can be observed that lexically similar languages boost the model's performance, leading to an improved cross-lingual transfer for the SelectNoise model. For example, languages like Kashmiri (kas), which have the lowest similarity, exhibit the lowest performance, whereas Chhattisgarhi (hne), with the highest lexical

ELRLs	Data size	BLEU	chrF
Hne	997	19.5	49.6
	6000	**20.3**	**50.3**
Mai	997	11.9	42.4
	6000	**12.4**	**43.2**
Npi	997	6.7	33.6
	6000	**7.2**	**33.8**

Table 7.21: SelectNoise Model performance with larger monolingual data

similarity, demonstrates the highest performance.

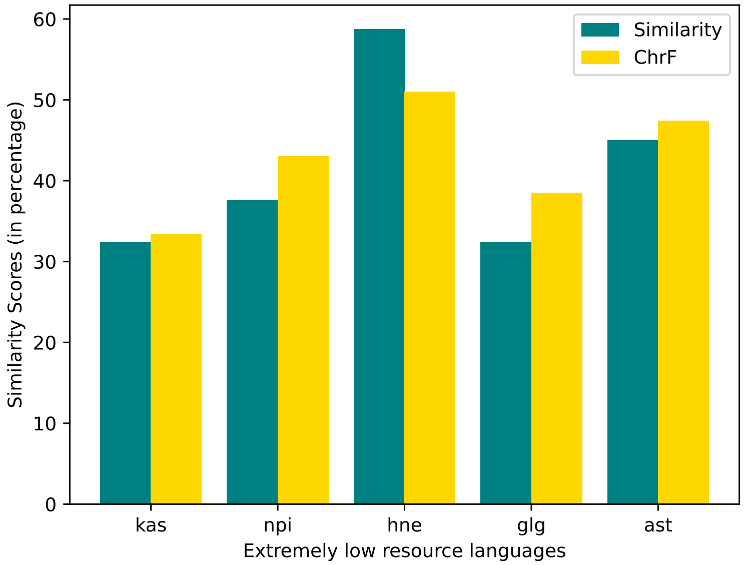


Figure 7.13: Language similarity vs. Performance.

Performance with Less Related Languages: We evaluate the zero-shot translation performance of Vanilla NMT and proposed SelectNoise models with two relatively less lexically similar ELRLs. These two languages belong to distinct language families, namely Bodo (Sino-Tibetan) and Tamil (Dravidian). Bodo has Devanagari script, while Tamil employs script conversion to match HRL (Hindi) script. The results are reported in Table 7.22. It is observed that the performance gain is minimal due to the dissimilarity of ELRLs with the corresponding HRL.

Performance for HRLs: Table 7.23 analyzes the performance of the proposed model for HRLs across both language families. It demonstrates comparable perfor-

Language	Model	BLEU	chrF
Bodo	Vanilla NMT	2.4	18.2
	SelectNoise	**2.7**	**18.7**
Tamil	Vanilla NMT	0.6	11.7
	SelectNoise	**0.9**	**13.3**

Table 7.22: Zero-shot translation performance of Vanilla NMT vs. SelectNoise on less related LRLs with HRL (Hindi)

Models	**Evaluation Metrics**			
	BLEU	**chrF**	**BLEURT**	**COMET**
Hindi HRL				
Vanilla NMT	33.4	60.2	0.724	0.868
Random Char Noise	33.0	59.5	0.722	0.865
SelectNoise	**34.2**	**60.5**	**0.726**	**0.869**
Spanish HRL				
Vanilla NMT	21.5	53.5	0.695	0.810
Random Char Noise	21.7	53.0	0.689	0.806
SelectNoise	**21.3**	**53.1**	**0.689**	**0.869**

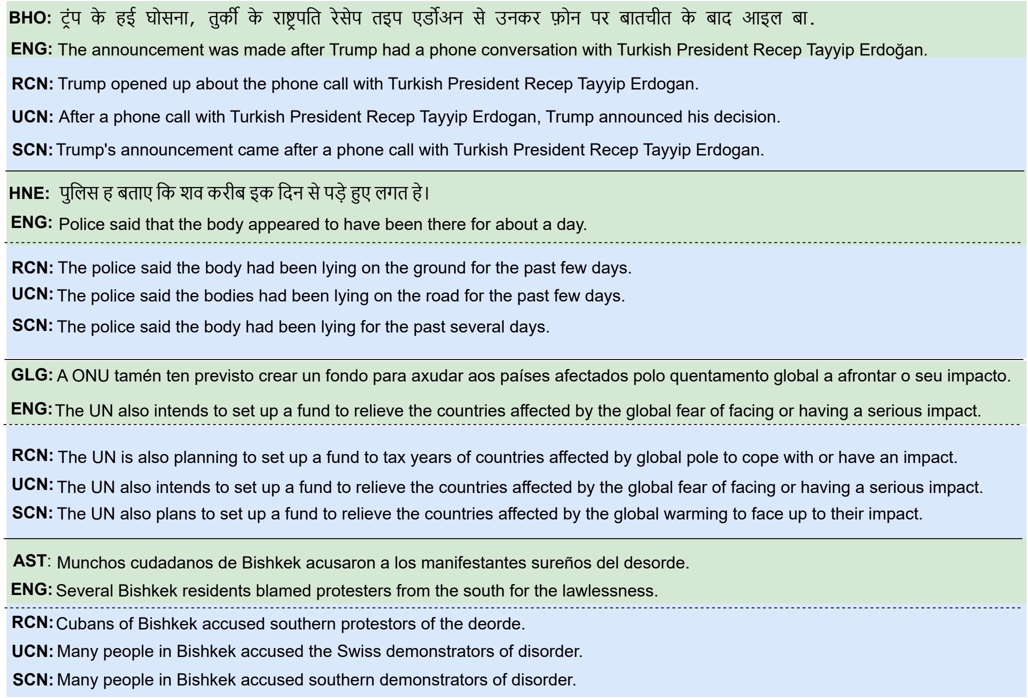
Table 7.23: Comparative performance for HRLs across both Indo-Aryan and Romance families.

mance with the vanilla NMT model for HRLs while boosting the performance of ELRLs. This highlights the effectiveness of the proposed model in handling both HRLs and ELRLs.

Sample Translations: Fig. 7.14 presents random sample generations/translations from Random Character Noise, SelectNoise and Supervised Character Noise injection models. It can be observed that the translation quality for the proposed SelectNoise model is much better than the baseline models.

7.4.4 Summary

This study presents an effective unsupervised approach, SelectNoise, for cross-lingual transfer from HRLs to closely related ELRLs through systematic character noise augmentation. The approach involves extracting selective noise augmentation candidates using BPE merge operations and edit operations. Furthermore, different sampling techniques are explored during the noise augmentation to ensure diverse candidate sampling. The model required only a small (1K example) amount of monolingual data in ELRLs. The proposed model consistently outperformed strong



BHO: ट्रंप के हई घोसना, तुर्की के राष्ट्रपति रेसेप तइप एर्डोअन से उनकर फ़ोन पर बातचीत के बाद आइल बा.

ENG: The announcement was made after Trump had a phone conversation with Turkish President Recep Tayyip Erdoğan.

RCN: Trump opened up about the phone call with Turkish President Recep Tayyip Erdogan.

UCN: After a phone call with Turkish President Recep Tayyip Erdogan, Trump announced his decision.

SCN: Trump's announcement came after a phone call with Turkish President Recep Tayyip Erdogan.

HNE: पुलिस ह बताए कि शव करीब इक दिन से पड़े हुए लगत हे।

ENG: Police said that the body appeared to have been there for about a day.

RCN: The police said the body had been lying on the ground for the past few days.

UCN: The police said the bodies had been lying on the road for the past few days.

SCN: The police said the body had been lying for the past several days.

GLG: A ONU tamén ten previsto crear un fondo para axudar aos países afectados polo quentamento global a afrontar o seu impacto.

ENG: The UN also intends to set up a fund to relieve the countries affected by the global fear of facing or having a serious impact.

RCN: The UN is also planning to set up a fund to tax years of countries affected by global pole to cope with or have an impact.

UCN: The UN also intends to set up a fund to relieve the countries affected by the global fear of facing or having a serious impact.

SCN: The UN also plans to set up a fund to relieve the countries affected by the global warming to face up to their impact.

AST: Munchos cudadanos de Bishkek acusaron a los manifestantes sureños del desorde.

ENG: Several Bishkek residents blamed protesters from the south for the lawlessness.

RCN: Cubans of Bishkek accused southern protestors of the deorde.

UCN: Many people in Bishkek accused the Swiss demonstrators of disorder.

SCN: Many people in Bishkek accused southern demonstrators of disorder.

Figure 7.14: Sample translations from various models for ELRLs to English MT direction. RCN: Random Character Noise injection model, UCN: Unsupervised Character Noise injection model (i.e., SelectNoise model) and SCN: Supervised Character Noise injection model.

baselines across 12 ELRLs from two diverse language families in the ELRLs-to-English MT task. The cumulative gain is 11.3% (chrF) over vanilla NMT. Further, the model demonstrated comparative performance to a supervised noise augmentation model.

7.5 Conclusion

In this chapter, we have presented two novel modeling techniques to enable and improve zero-shot ELRLs to English machine translation. These models are powered by a noise augmentation-based approach, acting as a regularizer to enhance the model's robustness against lexical variations. This, in turn, results in more effective cross-lingual transfer from HRL to closely related ELRLs. We have proposed two types of noise augmentation techniques: (1) The CharSpan model, which introduces random character-span noise augmentation and requires no additional learning resources for ELRLs. It is highly scalable. (2) The SelectNoise model applies more systematic and linguistically informed character noise augmentation. This model requires a small amount of monolingual data (1K) in ELRLs. Both models have different applicability: if no monolingual data is available, the CharSpan model is preferred, while if there is a small amount of monolingual data, the SelectNoise model is recommended. These models represent a significant advancement in the field of machine translation for ELRLs and are recognized as state-of-the-art models.

7.6 Insights, Limitations and Future Work

Insights: We have conducted several ablation experiments to ensure that the proposed design choices result in the best performance. Furthermore, our analysis indicates that the character-span-based model enhances the performance of languages that are less similar or more distant from HRLs. Additionally, it is important to select lexically similar languages HRLs in the multilingual training setup. Finally, we explore a multilingual setup in which multiple HRLs are trained together, resulting in a performance boost and scale coverage for ELRs. Our model performs equally well with a vocabulary that is learned with clean data. This provides scalability for utilizing PLMs, which typically have a fixed vocabulary.

Limitations: The current work addresses only transfer from related LRLs to English. It still remains to be investigated if noise augmentation is beneficial for translation from English to extremely low-resource languages. We assume that the related languages also use the same script or scripts that can be easily mapped/transliterated to each other. This method might not be effective for transfer between related languages that are not lexically similar or written in very different scripts, e.g., Hindi is written in the Devanagari script, while Sindhi is written in the Perso-Arabic script.

Future Work: In the future, we plan to extend the proposed models to ELRLs MT task and other NLG tasks. Additionally, we will explore modeling approaches for ELRLs that are not closely related to HRL or have different scripts. Improved modeling is needed to transform the language-specific features/aspects.

Chapter 8

Conclusion and Future Directions

Though Chapters 3–7 each contain their own conclusions, in this section, I will provide the overall conclusion and key takeaway to the readers from this thesis.

Natural language generation (NLG) is a well-explored research space where human-like text is generated given input context. NLG excels in personalization, automation, consistency, and multilingual versatility, transforming information into engaging stories. However, the extension of the modern NLG model is limited to three frequently occurring scenarios: (i) diverse text generation, (ii) text generation with limited context, and (iii) text generation with limited data/supervision for low-resource languages. In this thesis, I have focused on advancing Deep Learning based NLG modeling by mitigating these limitations. The proposed modeling approaches demonstrate impactful improvement and better suitability in real-life deployment. The conclusion provides a holistic summary of the thesis, key ideas and suggestions, and the value of the research below:

8.1 Summary of Contributions

We first address the task of distractor generation, i.e., generating multiple incorrect options given Multiple-Choice Question (MCQ) reading comprehension, i.e., input triplet ⟨ `passage, question, and correct answer` ⟩. With this objective in mind, we designed a semantic decoupling and hierarchical multi-decoder-based model that decouples the input context on the encoder side and employs interconnected multiple decoders to generate diverse distractors. This overcame the limitation of the existing models and generated distractors that were semantically not similar to the answer in the context of the question and exhibited lexical diversity among themselves. We evaluate the model performance with DG [GBL+19] and DG++ (prepared by us) datasets with 7 evaluation metrics. Our model outperforms the baseline models.

Our next model focused on addressing the limited context problem in extending NLG modeling to the personalized query auto-completions task. It is the task of generating top completions based on session and prefix inputs. The performance of existing models was limited for short and unseen prefixes due to a lack of relevant context within prefixes and in the session. However, the modeling with a retrieval-argument generation (RAG) framework yielded improved performance, especially for short and unseen prefixes. In this approach, external context in RAG was obtained from the traditional Trie model—an aspect not explored previously, and we were the first to leverage both the session and trie's knowledge. The evaluation was conducted with two real click-to-query datasets, namely Bing and AOL. On average, our model achieved a huge 57% and 14% boost in Mean Reciprocal Rank (MRR) compared to the popular trie-based lookup and the strong BART-based baseline methods, respectively.

The final research direction in this thesis extended NLG technology to many LRLs characterized by limited labeled data. Cross-lingual modeling was explored to transfer supervised signals from HRL to LRLs. However, zero-shot generation in LRLs presented an additional challenge—the catastrophic forgetting (CF) problem [XCR+21], where the generated text was either fine-tuning HRL or code-mixed with HRL and LRLs. To address this, an unsupervised adaptive training-based approach was proposed, which generates zero-shot, well-formed text in LRLs. This adaptive training required a small monolingual dataset (11k examples). The effectiveness of this model was tested across four NLG tasks and two LRLs. In the subsequent model, these findings were incorporated, with a focus on improving cross-lingual transfer signals. The improvement was achieved by employing meta-learning (i.e., MAML) and language clustering, resulting in more uniform cross-lingual supervision transfer to LRLs, even for less similar LRLs with HRL. The enhanced supervision significantly boosted the model's performance. To the best of our knowledge, this was the first study to apply meta-learning for zero-shot cross-lingual generation. The model underwent evaluation with two NLG tasks, 30 languages, and 5 public datasets, consistently outperforming a strong baseline. Finally, to push the boundaries, we proposed a modeling approach to enable language technology for extremely LRLs to English machine translation tasks. This was achieved through noise augmentation based on two approaches: random char-span noise augmentation (CharSpan) and systematic linguistically inspired character noise augmentation (SelectNoise). These approaches were evaluated across a large number of LRLs spanning different typologically diverse language families. Across all ELRLs and families, the CharSpan and SelectNoise models achieved gains of 9.46% and 11.31%, respectively, over the vanilla neural machine translation model [SHB16b].

8.2 Key Ideas and Suggestions

Based on the many failed and limited successful explorations with this thesis, I summarize several key take-away ideas from the thesis and our suggestions below:

Advancing Frontier of NLG with Constraints: The recurring theme of this part of the thesis is diverse text generation and text generation with limited context. The semantic alignment of diverse multiple outputs with each other is a key point to keep in mind. It is a vulnerable modeling point where more semantic alignment leads to more lexical similarity as well, and less semantic alignment leads to different semantics/meanings. On the other hand, the limited context problem is sensitive to the availability of external knowledge; the relevance and size of external knowledge directly impact the model performance in the RAG framework. If the curation of relevant knowledge matches the input context or is in the same domain, it boosts the performance. However, if the domain shift occurs, it can hamper the performance. Similarly, too little external context does not contribute to learning, or too much distracts the learning. Overall, the deep-leaning NLG models are sensitive to different aspects of modeling attributes and are most often determined empirically.

Low-resource Language Generation: The zero-shot cross-lingual modeling should be explored more exhaustively; although it is evolving, it has the potential to increase language coverage for NLP technology, benefiting the general population. Effective cross-lingual transfer requires more sophisticated modeling, specifically for NLG tasks. I believe the inclusion of different granular aspects like semantic, syntactic, and interlingual can pave the way forward. With the emergence of large multilingual language models, supervision transfer becomes more reliable, as these models represent all languages in a common latent representation space, being aware of the word/phrase/sentence semantics in different languages. There are many dialects across the globe, and more are emerging with time; enabling technology for dialects seems feasible, as supervision transfer is reasonable from closely related resources without the need for large learning resources for dialects. Two such modeling approaches we presented are CharSpan and SelectNoise. Overall, with recent advancements, this is an exciting time to do research in multilingual or low-resource NLP. With cross-lingual/multilingual/low-resource modeling, replicating the capabilities of *Bible fish* seems feasible.

8.3 The Value of Research

In this thesis, I have endeavored to extend natural language generation modeling. However, given the rapid advancement in generative AI, it's natural to wonder: *How much of this snapshot will be relevant a few years from now?*

I believe that the value of research lies not only in whether a particular technique is used in the future – in fact, almost all research will eventually be outdated and will pave the way for newer techniques and methods. Research is incremental work; the progress is made by standing on the shoulders of giants[1] — building on the foundations laid by earlier researchers. However, the challenges of diverse text generation and limited context may persist in various real-life applications. The proposed approaches are likely to remain effective in some form. The low-resource generation language holds more value, as the modeling is done in a zero-shot setting, ensuring scalability; these ideas may persist longer, as modern large language models often exhibit sub-optimal performance for low-resource languages (LRLs) [AHO+23]. Moreover, the proposed fine-tuning-based models explored in the thesis provide more controllability and interpretability compared to the unstructured modern prompting methods. I hope this thesis will inspire the reader to contribute to the collective understanding, which will be passed from generation to generation of researchers.

Lastly, ideas from the 80s and 90s, such as backpropagation, active learning, meta-learning, and many more, have resurfaced and become integral parts of cutting-edge modern systems. The current wave of large language models may take a tidal turn; however, some of our proposed techniques will still be there, and they may have something new to offer.

8.4 Future Directions

Due to the current advancement in generative NLP, the utilization and extension of ideas from this thesis are exciting. Here, we point to a few future research directions:

[1] https://en.wikipedia.org/wiki/Standing_on_the_shoulders_of_giants

- **Unified Modeling for Diverse Text Generation:** One interesting direction is to incorporate a diversifying module within the neural network. The diversifying modules should be application-agnostic and consider application-specific constraints. Furthermore, this should be a *plug-and-play* module for any neural network or large language model, enabling the generation of a diverse and arbitrary number of outputs. We have made a similar effort in [EMKD23] for diverse headline-generation applications. However, unified exploration is left for future work.

- **Advancing the RAG Modeling:** The scenario of limited context holds substantial relevance across numerous real-life applications, encompassing open-ended question answering, search queries, and providing relevant citations for generated output with large language models, among others. The RAG type of modeling is a promising direction where external knowledge is augmented from the web, databases, knowledge graphs, or even generated text from LLMs to overcome limited context issues. The recent advances in RAG modeling [LPP+20, WWS+22, WWS+22] have pushed the model's capabilities. External knowledge can be obtained using well-established approaches in information retrieval [Man09]. I believe this direction has to be explored more extensively in the future.

- **Language Technology for Next 7000+ Languages:** The research efforts should be directed toward developing a single, unified, and scalable modeling framework capable of addressing numerous NLP applications across 7000+ spoken languages. With the emergence of a large language model, the goal seems tractable. As a starting point, the creation of a large-scale multilingual NLP benchmark, akin to [HRS+20, AHO+23, CSW+22], should be prioritized, where the task and language coverage should be on a larger scale. This will help to push and track the progress of NLP research. Although I believe the model performance of the benchmark is not a true reflection of modeling capabilities in real-life applications, still, benchmarking will surely help bridge this gap.

- **Modeling Towards Multilinguality:** While this thesis delves into adaptive training, meta-learning, and noise augmentation for cross-lingual modeling, other trending directions are worth exploring: active learning [LWLH13], prompting [QWDC22], multi-tasking [GDA+21], etc. A more sophisticated modeling approach could be considered, for instance, learning a transformation function f from LRL to HRL. f can be viewed as an interlingua/unified space that accounts for linguistic, structural, and other typological features and facilitates better cross-lingual transfer. Overall, more such directions should be explored to improve cross-lingual transfer, enhancing performance collectively for a larger set of low-resource languages and NLP applications.

- **Evaluation of Multilingual NLG:** Evaluating NLG models is challenging due to the lack of reliable automated metrics and unbiased human evaluations [GCS23]. This challenge is more pronounced when dealing with multilingual NLG models, given the scarcity of linguistic tools and resources across languages. This underscores a critical research area that requires immediate attention and innovative solutions.

- **Evaluation without Reference:** Creating an evaluation benchmark for all languages and NLP tasks may not be feasible as new NLP tasks are frequently formulated. In light of this, there should be an effort to develop modeling techniques that allow evaluation without the need for gold reference evaluation data. This approach is explored in the machine translation task, which covers 1000 languages [SBF+22], providing a foundation for future research to explore similar directions.

References

[ACDGA12] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. 149

[AHO^{+}23] Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, et al. Mega: Multilingual evaluation of generative ai. *arXiv preprint arXiv:2303.12528*, 2023. 2, 161, 162

[AJF19] Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 122

[ALAC18] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*, April 2018. 125

[ARY20a] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics. xviii, 100, 109, 115

[ARY$^+$20b] Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online, July 2020. Association for Computational Linguistics. 125

[AS22] Noëmi Aepli and Rico Sennrich. Improving zero-shot cross-lingual transfer between closely related languages by injecting character-level noise. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4074–4083, Dublin, Ireland, May 2022. Association for Computational Linguistics. xxi, 124, 126, 128, 131, 134, 145, 146, 148

[ASC$^+$18] Ayana, Shi-qi Shen, Yun Chen, Cheng Yang, Zhi-yuan Liu, and Mao-song Sun. Zero-shot cross-lingual neural headline generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12):2319–2327, 2018. 101

[AWL$^+$21] Xiang Ao, Xiting Wang, Ling Luo, Ying Qiao, Qing He, and Xing Xie. PENS: A dataset and generic framework for personalized news headline generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 82–92, Online, August 2021. Association for Computational Linguistics. xv, 13

[BAN21] Damián Blasi, Antonios Anastasopoulos, and Graham Neubig. Systematic inequalities in language technology performance across the world's languages. *arXiv preprint arXiv:2110.06733*, 2021. 100, 101

[BBC90] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Citeseer, 1990. 100

[Ben11] Emily M Bender. On achieving and evaluating language-independence in nlp. *Linguistic Issues in Language Technology*, 6, 2011. 2, 122

[Ben19] Emily Bender. The# benderrule: On naming the languages we study and why it matters. *The Gradient*, 14, 2019. 1, 99, 121

[BJMM20] Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 522–534, 2020. 102

[BMD23] Maharaj Brahma, Kaushal Kumar Maurya, and Maunendra Sankar Desarkar. Selectnoise: Unsupervised noise injection to enable zero-shot machine translation for extremely low-resource languages. In *Findings of EMNLP*, 2023. 8, 124

[BMH$^+$22] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022. 53, 57

[BMM11] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. Query suggestions in the absence of query logs. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 795–804, 2011. 56

[BMR$^+$20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 18, 23, 26

[BPS$^+$12] Francesco Bonchi, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. Efficient query recommendations in the long tail via center-piece subgraphs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 345–354, 2012. 56

[BSP23] Verena Blaschke, Hinrich Schütze, and Barbara Plank. Does manipulating tokenization aid cross-lingual transfer? a study on POS tagging for non-standardized languages. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 40–54, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. 126

[Bur10] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581, 2010. 57

[BYK11] Ziv Bar-Yossef and Naama Kraus. Context-sensitive query autocompletion. In *Proceedings of the 20th international conference on World wide web*, pages 107–116, 2011. 56, 65

[CBOK06] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Reevaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006. 44

[CCB16] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany, August 2016. Association for Computational Linguistics. 126

[CCC$^+$20a] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. In *Transactions of the Association of Computational Linguistics*, 2020. xviii, 89, 115

[CCC$^+$20b] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020. 109, 110

[CDR16] Fei Cai and Maarten De Rijke. *Query auto completion in information retrieval.* Universiteit van Amsterdam [Host], 2016. 53

[CDW$^+$20a] Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. Cross-lingual natural language generation via pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7570–7577, Apr. 2020. 79, 82, 85, 111

[CDW$^+$20b] Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7570–7577, 2020. 101, 108

[CjCÇ$^+$22] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022. 121, 125, 129, 132, 147, 149

[CjEF17] Marta R. Costa-jussà, Carlos Escolano, and José A. R. Fonollosa. Byte-based neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 154–158, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. 126

[CK18] Ryan Cotterell and Julia Kreutzer. Explaining and generalizing back-translation through wake-sleep. *arXiv preprint arXiv:1806.04402*, 2018. 126

[CKG$^+$20] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. 77

[CLDR14] Fei Cai, Shangsong Liang, and Maarten De Rijke. Time-sensitive personalized query auto-completion. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1599–1608, 2014. 56

[CS18] Dhawaleswar Rao Ch and Sujan Kumar Saha. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 2018. 27, 29

[CSW+22] Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiaze Chen, Hao Zhou, and Lei Li. MTG: A benchmark suite for multilingual text generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2508–2527, Seattle, United States, July 2022. Association for Computational Linguistics. 162

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 16, 23, 28, 44, 84

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 82

[DH13] Matthew S Dryer and Martin Haspelmath. The world atlas of language structures online. 2013. 104

[DRAF17] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1747–1756, 2017. 54, 57

[DRK+21] Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676, 2021. 2

[DYZ+19] Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy, July 2019. Association for Computational Linguistics. 79, 101

[EMKD23] Venkatesh E, Kaushal Kumar Maurya, Deepak Kumar, and Maunendra Sankar Desarkar. Divhsk: Diverse headline generation using self-attention based keyword selection. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, July 2023. Association for Computational Linguistics. 2, 52, 162

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 10, 100, 102, 103

[FBS+21] Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48, 2021. 125

[FPLT14] Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2, 2014. 44

[Gag94] Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, feb 1994. 126

[GAG+17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. 20

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 18

[GBDG19] Rohit Gupta, Laurent Besacier, Marc Dymetman, and Matthias Gallé. Character-based nmt with transformer. *CoRR, abs/1911.04997.*, 2019. 126

[GBL+19] Yifan Gao, Lidong Bing, Piji Li, Irwin King, and Michael R Lyu. Generating distractors for reading comprehension questions from real examinations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6423–6430, 2019. xv, 4, 28, 30, 31, 38, 39, 42, 43, 44, 46, 50, 158

[GCS23] Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166, 2023. 163

[GDA$^+$21] Koustava Goswami, Sourav Dutta, Haytham Assem, Theodorus Fransen, and John P. McCrae. Cross-lingual sentence embedding using multi-task learning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9099–9113, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. 162

[GGS$^+$20] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November 2020. Association for Computational Linguistics. 74

[GHZ$^+$19] Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6251–6256, Hong Kong, China, November 2019. Association for Computational Linguistics. 102

[GKK$^+$16] Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P Bigham, and Emma Brunskill. Questimator: Generating knowledge assessments for arbitrary topics. In *IJCAI-16: Proceedings of the AAAI Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016. 30

[Goo] Google-2018. The wordpiece algorithm in open source bert. https://github.com/ google-research/bert/blob/master/tokenization.py#L335-L358. Retrieved on 11/01/2023. 126

[Goo77] Hubbard C Goodrich. Distractor efficiency in foreign language testing. *Tesol Quarterly*, pages 69–78, 1977. 27

[GSFP21] Xavier Garcia, Aditya Siddhant, Orhan Firat, and Ankur Parikh. Harnessing multilinguality in unsupervised machine translation for rare languages. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1126–1137, Online, June 2021. Association for Computational Linguistics. 122

[GWC+18] Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. 100

[GZW+19] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, 2019. 126

[HBFC19] Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *CoRR*, abs/1904.09751, 2019. 145

[HBI+21] Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August 2021. Association for Computational Linguistics. xv, xviii, 13, 108, 114

[HCO03] Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, 54(7):638–649, 2003. 56

[HFH17] Harald Hammarström, Robert Forkel, and Martin Haspelmath. Glottolog 3.0 (max planck institute for the science of human history, jena), 2017. 104

[HGJ+19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. 97

[HMRS14] Kajta Hofmann, Bhaskar Mitra, Filip Radlinski, and Milad Shokouhi. An eye-tracking study of user interactions with query auto completion. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 549–558, 2014. 56

[HO13] Bo-June Hsu and Giuseppe Ottaviano. Space-efficient data structures for top-k completion. In *Proceedings of the 22nd international conference on World Wide Web*, pages 583–594, 2013. 56

[HRS+20] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080, 2020. 25, 77, 100, 102, 106, 162

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 27

[HS16] Jennifer Hill and Rahul Simha. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30, 2016. 30

[IK19] Kango Iwama and Yoshinobu Kano. Multiple news headlines generation using page metadata. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 101–105, Tokyo, Japan, October–November 2019. Association for Computational Linguistics. 87

[Jai22] Shashank Mohan Jain. Hugging face. In *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*, pages 51–67. Springer, 2022. 51

[JCL+20] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 128

[JKCC14] Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 445–454, 2014. 56

[JL17] Shu Jiang and John Lee. Distractor generation for chinese fill-in-the-blank items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148, 2017. 30

[JSB+20] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July 2020. Association for Computational Linguistics. 1, 2, 99, 121

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 149

[KJM+19a] Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. Cross-lingual training for automatic question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872, Florence, Italy, July 2019. Association for Computational Linguistics. 79

[KJM+19b] Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. Cross-lingual training for automatic question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872, Florence, Italy, July 2019. Association for Computational Linguistics. 101

[KK17] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Ma-*

chine Translation, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. 122

[KKD+17] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. 45

[KLEG19] Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China, November 2019. Association for Computational Linguistics. 126

[KLJ+20] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*, 2020. 53, 57

[KMP+21] Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. Exploiting language relatedness for low web-resource language model adaptation: An Indic languages study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1312–1323, Online, August 2021. Association for Computational Linguistics. 126

[KR18] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. 21, 126

[KSB+22] Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A. Smith, and Daniel Weld. GENIE: Toward reproducible and standardized human evaluation for text generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11444–11458, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. 17

[Kun20] Anoop Kunchukuttan. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020. 135

[KZS+15] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 100

[LCDR18] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. April 2018. 125

[LCH17] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378, 10 2017. 126

[LCL+19] Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy, July 2019. Association for Computational Linguistics. 101

[LD09] Alon Lavie and Michael J Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23(2-3):105–115, 2009. 16, 44

[LDCM20a] Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online, November 2020. Association for Computational Linguistics. xviii, 108, 115

[LDCM20b] Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online, November 2020. Association for Computational Linguistics. 90

[LF20] Jindřich Libovický and Alexander Fraser. Towards reasonably-sized character-level transformer NMT by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579, Online, November 2020. Association for Computational Linguistics. 126

[LGG$^+$20a] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. Pre-training via paraphrasing, 2020. 101, 110

[LGG$^+$20b] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida I. Wang, and Luke Zettlemoyer. Pre-training via paraphrasing. *CoRR*, abs/2006.15020, 2020. 79

[LGG$^+$20c] 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. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. xx, 78, 80, 81, 82

[LGG$^+$20d] 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. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. 25, 26

[Lin04a] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 15, 44, 85

[Lin04b] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. 111

[LLG+19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 24, 26, 65

[LLG+20a] 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. 77

[LLG+20b] 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. 80, 125

[LLS+16] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas, November 2016. 44

[LML+17] Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain, April 2017. Association for Computational Linguistics. 104

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 23, 126

[LOR+20a] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online, July 2020. Association for Computational Linguistics. xviii, 89, 116

[LOR+20b] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online, July 2020. Association for Computational Linguistics. 109, 110

[LOYS19] Guokun Lai, Barlas Oguz, Yiming Yang, and Veselin Stoyanov. Bridging the domain gap in cross-lingual document classification. *CoRR*, abs/1909.07009, 2019. 100

[LPM15] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. 43, 46

[LPP+20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. 6, 75, 162

[LSF22] Jindřich Libovický, Helmut Schmid, and Alexander Fraser. Why don't people use character-level machine translation? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2470–2485, Dublin, Ireland, May 2022. Association for Computational Linguistics. 126

[LWLH13] Shoushan Li, Rong Wang, Huanhuan Liu, and Chu-Ren Huang. Active learning for cross-lingual sentiment classification. In *Natural Language Processing and Chinese Computing: Second CCF Conference, NLPCC*

2013, Chongqing, China, November 15-19, 2013, Proceedings 2, pages 236–246. Springer, 2013. 162

[LXL+17] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. xv, 12, 42, 49

[LYD+18] Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C Lee Giles. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 284–290, 2018. 30

[LYW+17] Chen Liang, Xiao Yang, Drew Wham, Bart Pursel, Rebecca Passonneaur, and C Lee Giles. Distractor generation with generative adversarial nets for automatically creating fill-in-the-blank questions. In *Proceedings of the Knowledge Capture Conference*, page 33. ACM, 2017. 30

[M+03] Ruslan Mitkov et al. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*, pages 17–22, 2003. 30

[Man09] Christopher D Manning. *An introduction to information retrieval*. Cambridge university press, 2009. 162

[MBH17] David Maxwell, Peter Bailey, and David Hawking. Large-scale generative query autocompletion. In *Proceedings of the 22nd Australasian Document Computing Symposium*, pages 1–8, 2017. 56

[MC15] Bhaskar Mitra and Nick Craswell. Query auto-completion for rare prefixes. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1755–1758, 2015. 4, 55, 56, 57, 60

[MD20a] Kaushal Kumar Maurya and Maunendra Sankar Desarkar. In *Proceedings of the 29th ACM International Conference on Information amp;*

Knowledge Management, CIKM '20, page 1115–1124, New York, NY, USA, 2020. Association for Computing Machinery. 29

[MD20b] Kaushal Kumar Maurya and Maunendra Sankar Desarkar. Learning to distract: A hierarchical multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1115–1124, 2020. 6, 91

[MD22] Kaushal Maurya and Maunendra Desarkar. Meta-x$_{NLG}$: A meta-learning approach based on language clustering for zero-shot cross-lingual transfer and generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 269–284, Dublin, Ireland, May 2022. Association for Computational Linguistics. 7, 100, 101

[MDGA23] Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Manish Gupta, and Puneet Agrawal. trie-nlg: trie context augmentation to improve personalized query auto-completion for short and unseen prefixes. *Data Mining and Knowledge Discovery*, 37(6):2306–2329, 2023. 6

[MDK20] Kelly Marchisio, Kevin Duh, and Philipp Koehn. When does unsupervised machine translation work? In *Proceedings of the Fifth Conference on Machine Translation*, pages 571–583, Online, November 2020. Association for Computational Linguistics. 125

[MDKD21a] Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. Zmbart: An unsupervised cross-lingual transfer framework for language generation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2804–2818. Association for Computational Linguistics, 2021. xviii, 101, 108, 110, 113, 114

[MDKD21b] Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. ZmBART: An unsupervised cross-lingual transfer framework for language generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages

2804–2818, Online, August 2021. Association for Computational Linguistics. 7, 77

[Mel95] I. Dan Melamed. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Third Workshop on Very Large Corpora*, 1995. 130

[MHM21] Shikhar Murty, Tatsunori Hashimoto, and Christopher D Manning. Dreca: A general task augmentation strategy for few-shot natural language inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1113–1125, 2021. 100

[MJVG21] Rajarshee Mitra, Rhea Jain, Aditya Srikanth Veerubhotla, and Manish Gupta. Zero-shot multi-lingual interrogative question generation for" people also ask" at bing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3414–3422, 2021. 114

[MKD$^+$21] Meryem M'hamdi, Doo Soon Kim, Franck Dernoncourt, Trung Bui, Xiang Ren, and Jonathan May. X-METRA-ADA: Cross-lingual meta-transfer learning adaptation to natural language understanding and question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3617–3632, Online, June 2021. Association for Computational Linguistics. 102

[MKDK24] Kaushal Kumar Maurya, Rahul Kejriwal, Maunendra Sankar Desarkar, and Anoop Kunchukuttan. Charspan: Utilizing lexical similarity to enable zero-shot machine translation for extremely low-resource languages. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Malta, 2024. Association for Computational Linguistics. 8, 124

[MLP20] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. Using bert and bart for query suggestion. In *CIRCLE*, 2020. 54, 57, 65

[MNL17] Chaitanya Malaviya, Graham Neubig, and Patrick Littell. Learning language representations for typology prediction. In *Proceedings of the*

2017 Conference on Empirical Methods in Natural Language Processing, pages 2529–2535, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. 104

[MPR22] Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States, July 2022. Association for Computational Linguistics. 126

[MSRH14] Bhaskar Mitra, Milad Shokouhi, Filip Radlinski, and Katja Hofmann. On user interactions with query auto-completion. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1055–1058, 2014. 56

[MZC08] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 469–478, 2008. 56

[NBBA20] Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. Zero-shot cross-lingual transfer with meta learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4547–4562. Association for Computational Linguistics, 2020. 102

[NC17] Toan Q. Nguyen and David Chiang. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. 125

[NgYW$^+$19] Feng Nie, Jin ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. A simple recipe towards reducing hallucination in neural surface realisation. In *ACL*, 2019. 111

[OEB$^+$19] 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 *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019. 132

[OHB20] Arturo Oncevay, Barry Haddow, and Alexandra Birch. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online, November 2020. Association for Computational Linguistics. xxi, 104, 105, 118

[PC17] Dae Hoon Park and Rikio Chiba. A neural language model for query auto-completion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1189–1192, 2017. 57

[PCP$^+$21] Amy Pu, Hyung Won Chung, Ankur P Parikh, Sebastian Gehrmann, and Thibault Sellam. Learning compact metrics for mt. In *Proceedings of EMNLP*, 2021. 149

[PCT06] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*, pages 1–es, 2006. 61

[PE09] Juan Pino and Maxine Eskenazi. Semi-automatic generation of cloze question distractors effect of students' l1. In *International Workshop on Speech and Language Technology in Education*, 2009. 30

[PEV20] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online, July 2020. Association for Computational Linguistics. 126, 131, 148

[PHE08] Juan Pino, Michael Heilman, and Maxine Eskenazi. A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada*, pages 22–32, 2008. 30

[PKR+21] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online, April 2021. Association for Computational Linguistics. 97

[Pop15] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. 16, 132, 149

[PRWZ02a] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002. 15, 44

[PRWZ02b] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 85

[PRWZ02c] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. 111

[PRWZ02d] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 132, 149

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 45

[PTS22] Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–233, Dublin, Ireland, May 2022. Association for Computational Linguistics. 126, 131, 148

[QWDC22] Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1910–1923, 2022. 162

[Rap21] Reinhard Rapp. Similar language translation for Catalan, Portuguese and Spanish using Marian NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 292–298, Online, November 2021. 130, 147

[RDB$^+$22] Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162, 2022. xix, 130, 147

[RJG$^+$18] Corby Rosset, Damien Jose, Gargi Ghosh, Bhaskar Mitra, and Saurabh Tiwary. Optimizing query evaluations using reinforcement learning for web search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1193–1196, 2018. 66

[RL12] Vasile Rus and Mihai Lintean. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 157–162, 2012. 44

[RNS+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 23, 26

[RSFL20] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics. 16, 132

[RSR+20a] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. volume 21, pages 1–67, 2020. 24, 26

[RSR+20b] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 65

[Rud22] Sebastian Ruder. The State of Multilingual AI. http://ruder.io/state-of-multilingual-ai/, 2022. 122

[RVS22] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland, May 2022. Association for Computational Linguistics. 2, 122

[RZLL16a] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. xv, 14

[RZLL16b] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural*

Language Processing, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. 89

[RZLL16c] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. 109

[SAK13] Keisuke Sakaguchi, Yuki Arase, and Mamoru Komachi. Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2)*, pages 238–242, 2013. 30

[SBF+22] Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*, 2022. 122, 125, 132, 149, 163

[SBV+15a] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 553–562, New York, NY, USA, 2015. Association for Computing Machinery. 40, 43, 46

[SBV+15b] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*, pages 553–562, 2015. 61

[SCY+18] Shi-qi Shen, Yun Chen, Cheng Yang, Zhi-yuan Liu, and Mao-song Sun. Zero-shot cross-lingual neural headline generation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(12):2319–2327, December 2018. 79

[SDP20] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. 16, 132, 149

[SFA+22] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022. 18, 23, 26

[SH16] Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August 2016. 28

[SHB16a] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany, August 2016. Association for Computational Linguistics. 126, 131, 148

[SHB16b] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. 21, 126, 131, 147, 160

[Sho13] Milad Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference on Re-*

search and development in information retrieval, pages 103–112, 2013. 56, 57

[SL21] Uri Shaham and Omer Levy. Neural machine translation without embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 181–186, Online, June 2021. Association for Computational Linguistics. 126

[SMR08] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008. 16

[SMWH10] Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. Clustering query refinements by user intent. In *Proceedings of the 19th international conference on World wide web*, pages 841–850, 2010. 56

[SNW17] Matthias Sperber, Jan Niehues, and Alex Waibel. Toward robust neural machine translation for noisy input sequences. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 90–96, 2017. 126

[SR12] Milad Shokouhi and Kira Radinsky. Time-sensitive query autocompletion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 601–610, 2012. 56

[SSY05] Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68, 2005. 30

[TBC$^+$21] Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online, November 2021. Association for Computational Linguistics. 125

[TCZ19] Min Tang, Jiaran Cai, and Hankz Hankui Zhuo. Multi-matching network for multiple choice reading comprehension. *Proceddings of the AAAI, Honolulu*, 2019. 35, 37

[TKK+21] Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. Meta-learning for effective multi-task and multilingual modelling. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3600–3612. Association for Computational Linguistics, 2021. 102

[TSKK19a] Norio Takahashi, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Machine comprehension improves domain-specific Japanese predicate-argument structure analysis. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 98–104, Hong Kong, China, November 2019. Association for Computational Linguistics. 89

[TSKK19b] Norio Takahashi, Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Machine comprehension improves domain-specific Japanese predicate-argument structure analysis. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 98–104, Hong Kong, China, November 2019. Association for Computational Linguistics. 109

[VBL+22] Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. Overcoming catastrophic forgetting in zero-shot crosslingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. 15

[vdHYMS21] Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. Multilingual and cross-lingual document classification: A meta-learning approach. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1966–1976, Online, April 2021. Association for Computational Linguistics. 102

[VdVT19] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 77, 95

[vELR+22] Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara Rivera. Writing system and speaker metadata for 2,800+ language varieties. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5035–5046, 2022. 2

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 19, 20, 57, 127, 141

[WBGL16] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 44

[WBSG10] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010. 56

[Wen14] Etienne Wenger. *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge.* Morgan Kaufmann, 2014. 3

[WLG17] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark, September 2017. 27

[WLW+20] Qianhui Wu, Zijia Lin, Guoxin Wang, Hui Chen, Börje F Karlsson, Biqing Huang, and Chin-Yew Lin. Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9274–9281, 2020. 102

[WLX10a] Xiaojun Wan, Huiying Li, and Jianguo Xiao. Cross-language document summarization based on machine translation quality prediction.

In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden, July 2010. Association for Computational Linguistics. 79

[WLX10b] Xiaojun Wan, Huiying Li, and Jianguo Xiao. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, 2010. 101

[WPAN19a] Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*, 2019. 126

[WPAN19b] Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. Multilingual neural machine translation with soft decoupled encoding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 131

[WPDN18] Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. 126, 131, 148

[WWS$^+$22] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 162

[WZM$^+$18] Po-Wei Wang, Huan Zhang, Vijai Mohan, Inderjit S Dhillon, and J Zico Kolter. Realtime query completion via deep language models. In *eCOM@ SIGIR*, 2018. 57

[XCR$^+$21] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. 2, 5, 25, 26, 77, 105, 106, 107, 108, 159

[YJT+23] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. 2023. 1

[YSH+21] Nishant Yadav, Rajat Sen, Daniel N Hill, Arya Mazumdar, and Inderjit S Dhillon. Session-aware query auto-completion using extreme multi-label ranking. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3835–3844, 2021. 16, 57, 61, 64

[YTZ+20] Di Yin, Jiwei Tan, Zhe Zhang, Hongbo Deng, Shujian Huang, and Jiajun Chen. Learning to generate personalized query auto-completions via a multi-view multi-task attentive approach. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2998–3007, 2020. 54, 55, 57

[YZJZ20] Ming Yan, Hao Zhang, Di Jin, and Joey Tianyi Zhou. Multi-source meta transfer for low resource multiple-choice question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7331–7341, 2020. 100

[ZKW+20] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 16, 85

[ZLW20] Xiaorui Zhou, Senlin Luo, and Yunfang Wu. Co-attention hierarchical network: Generating coherent long distractors for reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9725–9732. AAAI Press, 2020. xv, 4, 30, 31, 43, 46, 50

[ZM14] Torsten Zesch and Oren Melamud. Automatic generation of challenging distractors using context-sensitive inference rules. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148, 2014. 30

[ZNDK18] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. 89

[ZWL+19] Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450, 2019. 126

[ZYMK16] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November 2016. Association for Computational Linguistics. 125

[ZZL+23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 1

[ZZS+18] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018. 57