

TOWARDS RELIABLE AND INCLUSIVE NATURAL LANGUAGE
GENERATION

Shiyue Zhang

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2023

Approved by:

Mohit Bansal

Asli Celikyilmaz

Snigdha Chaturvedi

Ido Dagan

Junier Oliva

© 2023
Shiyue Zhang
ALL RIGHTS RESERVED

ABSTRACT

Shiyue Zhang: Towards Reliable and Inclusive Natural Language Generation
(Under the direction of Mohit Bansal)

Natural language generation (NLG) is an important subfield of natural language processing (NLP) that produces natural language output. Despite notable advancements made by large-scale pre-trained language models in NLG, there remain several unresolved challenges. This thesis aims to enhance NLG from two significant aspects: reliability and inclusiveness.

For reliability, on the one hand, we introduce novel training objectives that improve the alignment of language generation models with desired model behaviors. To improve the answerability of model-generated questions, we use a question answering model to provide additional rewards to a question generation model, encouraging the production of more answerable questions. In addition, we propose to train language models with a mixture of forward and reverse cross-entropies, demonstrating that the resulting models yield better generated text without complex decoding strategies. On the other hand, we propose novel evaluation methods to assess the performance of NLG models accurately and comprehensively. By combining human and automatic evaluations, we strike a balance between reliability and reproducibility. We delve into the unexplored issue of unfaithfulness in extractive summaries and conclude that extractive summarization does not guarantee faithfulness.

For inclusiveness, we extend the coverage of NLG techniques to low-resource or endangered languages. We develop the first machine translation system for supporting translation between Cherokee, an endangered Native American language, and English, and we propose a roadmap for utilizing NLP to support language revitalization efforts. Additionally, we investigate the underrepresentation of low-resource languages during multilingual tokenization, a crucial data pre-

processing step in training multilingual NLG models, and we present best practices for training multilingual tokenizers.

Overall, this thesis works towards enhancing the trustworthiness of NLG models in practice and facilitating support for a more diverse range of languages worldwide.

To my parents.

“Hope I keep learning and growing as a researcher.”

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my advisor, Professor Mohit Bansal, for his support throughout my Ph.D. studies. Mohit has been a constant source of guidance, not only in my research endeavors but also in his genuine concern for my well-being, helping me navigate through challenging times. Thanks to his mentorship, my five years at UNC-Chapel Hill have been an enriching experience, one that I will forever cherish without any regrets.

I would like to thank the Bloomberg Data Science Ph.D. Fellowship for their generous support during the final two years of my Ph.D. studies. I also appreciate the exceptional internship experience I had at Bloomberg in the summer of 2022, which led to a full-time return offer, and I will be joining Bloomberg upon my graduation.

I would like to thank all my recommenders: Mohit Bansal, Dong Wang, Asli Celikyilmaz, Francisco Guzmán, and David Rosenberg. Their invaluable support and guidance have been instrumental throughout my Fellowship application and job search processes. I am truly fortunate to have such outstanding individuals in my corner advocating for me. They are my role models in my research career.

I would like to thank all my collaborators during my Ph.D. studies: Mohit Bansal, Peter Hase, Harry Xie, Benjamin Frey, Zineng Tang, Hyounghun Kim, Asli Celikyilmaz, Jianfeng Gao, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Francisco Guzman, Xiang Zhou, Yinuo Hu, Viji Sathy, A. T. Panter, Swarnadeep Saha, Derek Tam, Anisha Mascarenhas, Sarah Kwan, Colin Raffel, Shijie Wu, Ozan Irsoy, Steven Lu, Mark Dredze, and David Rosenberg.

I would like to thank my committee members, Mohit Bansal, Snigdha Chaturvedi, Asli Celikyilmaz, Ido Dagan, and Junier Oliva, for their constant support and valuable feedback in completing this dissertation.

I would like to thank the cohesive and diverse lab that I have been honored to be a part of, as well as all of my current and former lab mates. I have consistently felt a strong sense of respect and belonging in our lab, owing to the invaluable diversity that each individual brings. The support and willingness to help from my lab mates have cultivated an environment where collaboration thrives, empowering me to tackle challenges and accomplish achievements.

I would like to thank my friends at UNC-Chapel Hill, with special thanks to Peirong Liu, Yubo Zhang, and Yixin Nie. Their presence and companionship during every Chinese New Year have meant the world to me. Their friendship has provided me with a sense of belonging and has been a source of strength during both joyous and challenging times.

I would like to thank my friends in China, especially my college roommates and my life-long friends Yanan Zhang and Yuanyuan Wang. Among them, Yuanyuan holds a special place in my life as not only my closest friend but also a sister-like figure. Despite the physical distance, we have maintained regular communication, and her unwavering presence has been a constant source of comfort and support for me.

I would like to thank my partner, Xiang Zhou. I am truly blessed to meet him at UNC. His support and love have been the cornerstone of my achievements during my Ph.D. His kindness, creativity, and optimism have had a profound impact on me and constantly motivated me to become a better version of myself.

Lastly and most importantly, I would like to thank my parents for their constant support and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES	xiii
LIST OF FIGURES	xviii
CHAPTER 1: INTRODUCTION	1
1.1 Thesis Statement	6
1.2 Overview of Chapters	6
CHAPTER 2: ADDRESSING SEMANTIC DRIFT IN QUESTION GENERATION	7
2.1 Introduction.....	7
2.2 Background and Related Work	9
2.3 Question Generation	11
2.3.1 Base Model	11
2.3.2 Semantics-Reinforced Model.....	13
2.3.3 QA-Based QG Evaluation.....	15
2.4 Semi-Supervised Question Answering.....	16
2.4.1 Synthetic Data Generation	16
2.4.2 Synthetic Data Usage	17
2.5 Experiment Setup	18
2.5.1 Datasets.....	18
2.5.2 Evaluation Metrics	19
2.5.3 Implementation Details	20
2.6 Results	22
2.6.1 Question Generation	22

2.6.2	Semi-Supervised Question Answering	24
2.6.3	QG and QA Results with BERT	26
2.6.4	Examples	27
2.7	Conclusion	27
CHAPTER 3: TRAINING LANGUAGE MODELS BY MIXING FORWARD AND REVERSE CROSS-ENTROPIES		29
3.1	Introduction.....	29
3.2	Background and Related Work	32
3.2.1	Autoregressive Language Modeling	32
3.2.2	Language <i>Degeneration</i>	32
3.2.3	Objectives Beyond MLE.....	33
3.3	Methodology	35
3.3.1	MixCE.....	35
3.3.2	Optimization of Reverse CE	36
3.3.3	Connection to Pang and He (2021)	40
3.4	Experiments	41
3.4.1	Synthetic Experiments	41
3.4.2	GPT-2 Experiments.....	45
3.4.3	Robustness & Analysis	48
3.5	Implementation Details	51
3.5.1	Best η	51
3.5.2	Human Evaluation Details	52
3.5.3	Reproducibility.....	54
3.6	Conclusion	55
CHAPTER 4: SEMI-AUTOMATIC SUMMARY EVALUATION		58
4.1	Introduction.....	58
4.2	Background and Related Work	60

4.3	Our Method	63
4.3.1	Lite ² yramid	63
4.3.2	Lite ³ yramid	65
4.3.3	Lite ^{2.x} yramid	67
4.4	Evaluation	68
4.4.1	Correlation with Human Scores	68
4.4.2	Metrics for Comparison	69
4.4.3	Data	69
4.4.4	Models	70
4.5	Results	71
4.5.1	Human-Metric Correlation Results	71
4.5.2	Out-of-the-Box Generalization	74
4.5.3	Performance of Individual Modules	77
4.6	Implementation Details	78
4.6.1	PyrXSum	78
4.6.2	Experimental Details	81
4.7	Conclusion	82
CHAPTER 5: EXTRACTIVE IS NOT FAITHFUL: FAITHFULNESS EVALUA- TION FOR EXTRACTIVE SUMMARIZATION		83
5.1	Introduction	83
5.2	Broad Unfaithfulness Problems	87
5.3	Human Evaluation	90
5.3.1	Data	91
5.3.2	Setup	92
5.3.3	Results of Human Evaluation	93
5.4	Automatic Evaluation	94
5.4.1	Meta-evaluation Method	94

5.4.2	Existing Faithfulness Evaluation Metrics	96
5.4.3	A New Metric: ExtEval	97
5.4.4	Meta-Evaluation Results	99
5.5	Generalizability	101
5.6	Implementation Details	102
5.6.1	Human Evaluation Details	102
5.6.2	ExtEval Details	102
5.7	Conclusion	103
CHAPTER 6: CHEROKEE-ENGLISH MACHINE TRANSLATION AND BEYOND		109
6.1	Introduction	109
6.2	The Cherokee Language	115
6.2.1	History of the Cherokee People and Their Language	115
6.2.2	Cherokee Linguistics	117
6.3	ChrEn Dataset	119
6.3.1	Data Collection	119
6.3.2	Models	122
6.3.3	Results	128
6.3.4	Implementation Details	133
6.4	ChrEnTranslation System	140
6.4.1	System Description	140
6.4.2	Evaluation	145
6.5	Using NLP to Assist in Language Revitalization	149
6.5.1	Before Diving into NLP Research	149
6.5.2	NLP-Assisted Language Education	151
6.5.3	NLP Tools for Cherokee Language Processing	153
6.6	Conclusion	158

CHAPTER 7: LANGUAGE IMBALANCE IN MULTILINGUAL TOKENIZER TRAINING	159
7.1 Introduction.....	159
7.2 Background and Related Work	161
7.2.1 Tokenization Methods	161
7.2.2 Multilingual Tokenization.....	162
7.2.3 Analysis and Assessment of Tokenization	162
7.3 Bilingual Experiments	163
7.3.1 Experimental Setup	164
7.3.2 Intermediate Features	166
7.3.3 Translation Results	167
7.3.4 Ablations	170
7.4 Multilingual Experiments.....	174
7.4.1 Experiment Setup & Features	175
7.4.2 Results & Ablations	177
7.5 Implementation Details	179
7.6 Conclusion	180
CHAPTER 8: SUMMARY, LIMITATIONS, ETHICS, AND FUTURE WORK	182
8.1 Summary of Contributions.....	182
8.2 Limitations and Future Work.....	184
8.3 Ethical Considerations.....	186
REFERENCES	188

LIST OF TABLES

Table 2.1 – The performance of different QG models.	22
Table 2.2 – Pairwise human evaluation between our baseline and QPP&QAP multi-reward model.	22
Table 2.3 – The QA-based evaluation results for different QG systems. The two numbers of each item in this table are the EM/F1 scores. All results are the performance on our QA test set. “S” is short for “SQuAD”.	23
Table 2.4 – The effect of QAP-based synthetic data filter. We filter out the synthetic data with $QAP < \epsilon$. All results are the performance on our QA development set.	24
Table 2.5 – The results of our semi-supervised QA method using a BiDAF-QA model.	24
Table 2.6 – The comparison with the previous semi-supervised QA method. All results are the performance on the full development set of SQuAD, i.e., our QA test + development set.	25
Table 2.7 – The performance of our stronger BERT-QG models.	26
Table 2.8 – The results of our semi-supervised QA method using a stronger BERT-QA model.	26
Table 3.1 – Synthetic experimental results. Random (10%, 50%, 90%) randomly initializes \mathbf{M} and sets 10% or 50% or 90% of the probabilities to 0. WebText means initializing \mathbf{M} by the bigram occurrence in the WebText data. Gold refers to the results when $\mathbf{M}'=\mathbf{M}$. <i>avg. js</i> is our main metric, which represents the average JS divergence between \mathbf{M} and \mathbf{M}' (please see the definition of <i>avg. js</i> in text). Each number is a 5-seed average, and Table 3.2 shows the 95% confidence intervals of some experiments.	43
Table 3.2 – Synthetic experimental results with 95% confidence intervals. WebText means initializing \mathbf{M} by the bigram occurrence in the WebText data.	44
Table 3.3 – Unbiased sampling results of models finetuned by MLE or MixCE on three datasets. For all metrics, the closer to the human scores the better. Bold numbers are the ones that are closer to human scores in each setting. Each number is a 3-run average.	47

Table 3.4 – Top- p sampling results of the same models as Table 3.3. Since changing the decoding method will not affect perplexity, we report the selected best p instead.	47
Table 3.5 – Human evaluation results. The star (*) means significantly better ($p < 0.01$). The significance test is conducted following the bootstrap test setup (Efron and Tibshirani, 1994).....	48
Table 3.6 – Unbiased sampling results of GPT-2 small models finetuned by MLE or MixCE on three datasets of different training data sizes. All metrics are the closer to the human scores the better. Bold numbers are the ones that are closer to human scores in each setting.	48
Table 3.7 – Controlled mauve results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 3.3. Human scores are not 1 because sampling 10K fragments twice result in two different sets. Each number is a 3-run average.	49
Table 3.8 – Controlled coherence results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 3.3. Each number is a 3-run average.	49
Table 3.9 – Unbiased sampling text lengths of models finetuned by MLE or MixCE on three datasets. Length is computed by simply splitting text by whitespaces.	51
Table 3.10 –The selected best η of synthetic experiments reported in Table 3.1. The model section is based on avg. js.	52
Table 3.11 –The selected best η of GPT-2 experiments reported in Table 3.3. The model section is based on mauve (max length=512) on the dev set.	53
Table 3.12 –Inter-annotator agreement. The numbers are the portions of examples that have a 3-annotator agreement (all agree), a 2-annotator agreement (2 agree), or no agreement. E.g., 24% of examples used in human evaluation for Wiki-Text have an agreement among 3 annotators.	54
Table 4.1 – 5-fold (split by examples) cross-validation results. In each column, the bold numbers are the best and the <u>underline</u> numbers are the best out of automatic metrics. All Lite ² Pyramid-0 numbers are based on $f_{\text{nli}} = l^{3c}$, while all other numbers of our metrics are based on $f_{\text{nli}} = p^{2c}$	71

Table 4.2 – 5-fold (split by systems) cross-validation results. In each column, the bold numbers are the best and the <u>underline</u> numbers are the best out of automatic metrics. All Lite ² Pyramid-0 numbers are based on $f_{\text{nli}} = l^{3c}$. All other numbers of our metrics are based on $f_{\text{nli}} = p^{2c}$, except that those star* numbers are based on $f_{\text{nli}} = l^{2c}$.	75
Table 4.3 – Out-of-the-box generalization results. In each column, the bold numbers are the best and the <u>underline</u> numbers are the best out of automatic metrics.	76
Table 4.4 – The ROUGE-2 (R2) and gold Pyramid scores obtained by 10 systems on the 100 XSum testing examples.	78
Table 5.1 – All metric scores and the human Overall score for the 16 extractive systems on the 100 CNN/DM testing examples. The score of a system is the average score of 100 examples. ↓ means the scores are the lower the better.	97
Table 5.2 – Human-metric correlations. The negative sign (-) before metrics means that their scores are negated to retain the feature that the higher the scores are the more unfaithful the summaries are.	98
Table 5.3 – System-level and summary-level correlations. The negative sign (-) before metrics means that their scores are negated to retain the feature that the higher the scores are the unfaithful the summaries are.	100
Table 5.4 – Meta-evaluation results on SummEval (Fabbri et al., 2021). Method 1 refers to the meta-evaluation method used in Section 5.4.1, while Method 2 refers to the system-level correlation used by Fabbri et al. (2021). We negate ExEval to make higher scores mean more faithful.	101
Table 6.1 – An example from the development set of ChrEn. NMT denotes our RNN-NMT model.	109
Table 6.2 – The key statistics of our parallel and monolingual data. Note that “% Unseen unique English tokens” is in terms of the Train split, for example, 13.3% of unique English tokens in Dev are unseen in Train.	120
Table 6.3 – Parallel Data Sources.	123
Table 6.4 – Monolingual Data Sources.	124
Table 6.5 – Performance of our supervised/semi-supervised SMT/NMT systems. Bold numbers are our best out-of-domain systems together with Table 6.6, selected by performance on Out-dev. (±x) shows 95% confidence interval.	129

Table 6.6 – Performance of our transfer and multilingual learning systems. Bold numbers are our best in-domain systems together with Table 6.5, selected by the performance on Dev. ($\pm x$) shows the 95% confidence interval.	131
Table 6.7 – Human comparison between the translations generated from our NMT and SMT systems. If A vs. B, “Win” or “lose” means that the evaluator favors A or B. Systems IDs correspond to the IDs in Table 6.5.	133
Table 6.8 – The hyper-parameter settings of Supervised and Semi-supervised Cherokee-English NMT systems in Table 6.5 . Empty fields indicate that hyper-parameter is the same as the previous (left) system.	136
Table 6.9 – The hyper-parameter settings of Transferring Cherokee-English NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.	137
Table 6.10 –The hyper-parameter settings of Multilingual Cherokee-English NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.	138
Table 6.11 –The hyper-parameter settings of in-domain Supervised and Semi-supervised English-Cherokee NMT systems in Table 6.5. Empty fields indicate that hyper-parameter is the same as the previous (left) system.	138
Table 6.12 –The hyper-parameter settings of out-of-domain Supervised and Semi-supervised English-Cherokee NMT systems in Table 6.5. Empty fields indicate that hyper-parameter is the same as previous (left) system.	139
Table 6.13 –The hyper-parameter settings of Transferring English-Cherokee NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.	139
Table 6.14 –The hyper-parameter settings of Multilingual English-Cherokee NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.	140
Table 6.15 –Pearson correlation coefficients between QE and BLEU or between QE and human rating. “/ length” represents the normalization by output sentence length.	146
Table 6.16 –The performance of translation models.	146
Table 6.17 –Expert feedback. In each cell, the 3 numbers are the number of feedback received / average quality rating / Pearson correlation coefficient between quality rating and quality estimation.	149

Table 6.18 –OCR performance of two OCR tools on our evaluation sets. WER: word error rate, CER: character error rates. For both WER and CER, lower is better.	154
Table 6.19 –The ASR results of finetuned XLSR-53 (Conneau et al., 2020) models. WER: word error rate.	155
Table 6.20 –The alignment between subwords and gold morphemes.	156
Table 7.1 – 8 languages in our experiments. K/M/B stands for thousand/million/billion. Mono. stands for monolingual. Numbers are the number of sentences (pairs).	164
Table 7.2 – Translation results (spBLEU scores) of adding the non-Latin language’s characters to the vocabulary at English=100% (100%+char). For comparison, the 100% column shows the results before adding characters and the best column shows the best results out of all percentages.	174
Table 7.3 – Comparison of language sampling factors used in tokenizer or model training. All numbers are spBLEU. S is the exponential factor used in temperature sampling (see Section 7.2.2).....	179
Table 7.4 – Examples of how sentences in English, Indonesian, and Chinese are tokenized at different English percentages under our main bilingual setting (Section 7.3.1). The sentence is the first sentence of Flores101 devtest set. Subwords are separated by whitespaces, and unknown tokens are replaced by ‘<unk>’.	181

LIST OF FIGURES

Figure 2.1 – An examples of the “semantic drift” issue in Question Generation (“Gt” is short for “ground truth”).	8
Figure 2.2 – The architecture of our semantics-reinforced QG model.	13
Figure 2.3 – Two examples of where QPP and QAP improve in question quality evaluation.	14
Figure 2.4 – Semi-supervised QA: First, a trained QG model generates questions from new or existing paragraphs building up a synthetic QA dataset; Second, a data filter filters out low-QAP synthetic examples and augment the rest to human-labeled QA pairs; Lastly, the QA model is trained with the enlarged QA dataset.	16
Figure 2.5 – Some synthetic QA examples generated by our QG models.	28
Figure 3.1 – MixCE combines two complementary driving forces: reverse CE helps narrow the model distribution Q_θ down when it is broader than data distribution P , while forward CE helps broaden Q_θ out when it is narrower than P . Note that $\log P(x)$ is infinite when $P(x) = 0$. But in practice, we use $\log P(x) = \sum_t \log(P(x_t x_{<t}) + \epsilon)$ to avoid $\log 0$ and $\epsilon = 1e - 30$	30
Figure 3.2 – Forward CE only weakly penalizes the model Q_θ when it puts a small amount of probability mass onto $P(x)=0$ space. And the loss magnitude is much smaller than what we will get from reverse CE.	36
Figure 3.3 – The histograms of sequence-level and token-level negative log-likelihoods of human texts and model generations from GPT-2 large.	38
Figure 3.4 – The mauve scores obtained by MixCE-finetuned GPT-2 models on development sets with different max generation lengths and different η . Note that when $\eta = 1$, MixCE is equivalent to MLE. The x-axis is the mixing ratio η , and the y-axis refers to mauve scores with different max generation lengths. The 3 lines in each subplot show the results of GPT-2 models in different sizes. The 3 subplots in each row are the results of 3 datasets respectively. Unbiased sampling is used as the decoding method. Each dot is the average of 3 runs of sampling and the error bar shows the standard deviation of 3 runs.	56
Figure 3.5 – Human evaluation interface and a random example from our collected human annotations.	57

Figure 4.1 – The illustration of our metrics. This data example is from REALSumm (Bhandari et al., 2020) (we omit unnecessary content by ‘...’). For gold labels, ‘1’ stands ‘present’ and ‘0’ stands ‘not present’. Other scores are the 2-class entailment probabilities, $p^{2c}(e)$, from our finetuned NLI model.	63
Figure 4.2 – Lite ^{2.x} Pyramid curves and its comparison to replacing <i>random</i> sentences’ SCUs with STUs.	73
Figure 4.3 – Lite ^{2.x} Pyramid curves (for system-level correlations) and its comparison to replacing <i>random</i> sentences’ SCUs with STUs.	74
Figure 4.4 – The Amazon Mechanical Turk user interface for collecting human labels of SCUs’ presence.	79
Figure 5.1 – An example from CNN/DM (Hermann et al., 2015) testing set showing the first four types of unfaithfulness problems defined in section 5.2. The three summaries are generated by NeuSumm (Zhou et al., 2018a) Oracle (disco) (Xu et al., 2020a), and BERT+LSTM+PN+RL (Zhong et al., 2019), respectively. All extracted sentences or discourse units are <u>underlined</u> in the document. The problematic parts are bolded in the summary. The incorrect reference in the summary is marked with red , and the correct reference is marked with blue in the document. We replace non-relevant sentences with [...].	86
Figure 5.2 – Our typology of broad unfaithfulness problems in extractive summarization. ...	87
Figure 5.3 – The unfaithfulness error distributions of 16 extractive summarization systems...	92
Figure 5.4 – An example from CNN/DM (Hermann et al., 2015) testing set showing an <i>incomplete coreference</i> error. The summary is generated by BERT+LSTM+PN+RL (Zhong et al., 2019). All extracted sentences are <u>underlined</u> in the document. The word its in the summary is ambiguous. It can refer to PG&E or California Public Utilities Commission. The correct coreference should be PG&E in the document.	104
Figure 5.5 – An example from CNN/DM (Hermann et al., 2015) testing set showing an <i>incomplete discourse</i> error. The summary is generated by the Oracle (disco) (Xu et al., 2020a) extractive system. All extracted elementary discourse units are <u>underlined</u> in the document. The last summary sentence missed the “born in the u.s” part which may make people think the Busby girls is the first all-female quintuplets not only in US.	105

Figure 5.6 – An example from CNN/DM (Hermann et al., 2015) testing set showing a <i>other misleading information</i> error. The summary is generated by the HeterGraph (Wang et al., 2020b) extractive system. All extracted sentences are <u>underlined</u> in the document. If readers only read the summary, they may think the football players and pro wrestlers won the contest and ate 13 pounds of steak.	106
Figure 5.7 – An example of post-correction with ExtEval. In the original summary, <i>they</i> refers to <i>the vessel and crew</i> in the summary, but it only refers to <i>the crew</i> in the document. In the corrected summary, the automated program successfully replaces <i>they</i> with <i>the crew members</i> ’ though with a minor grammar issue.	107
Figure 5.8 – The interface for human annotation.	108
Figure 6.1 – Language family trees.	111
Figure 6.2 – The distributions of our parallel (Para.) and monolingual (Mono.) data over text types and dialects.	120
Figure 6.3 – A simple illustration of SMT and NMT.	125
Figure 6.4 – The four different ways we proposed to incorporate BERT representations into NMT models.	127
Figure 6.5 – The influence of the English monolingual data size on semi-supervised learning performance. The results are on Dev or Out-dev.	137
Figure 6.6 – Translation interface of our demonstration system. Note that “ᏓᏍᏉᏃᏍᏉᏃᏍᏉ DT.” is not a correct translation. See Figure 6.7 for the corrected translation by an expert.	142
Figure 6.7 – Two user feedback interfaces of our demonstration system. (b) shows the feedback given by an expert.	143
Figure 6.8 – Word alignment visualization and link to Cherokee-English Dictionary.	144
Figure 7.1 – Results of our main bilingual experiments. Marker shapes denote the language pairs; dash or solid lines represents out-of-English or into-English directions; colors are for each target language. E.g., --▲- (en-ta) denotes Tamil features (<i>Closeness to the character level</i> or <i>UNK rate</i>) or English to Tamil translation results (<i>spBLEU</i> or <i>chrF</i> scores); -▲ (ta-en) represents English features or Tamil to English translation results. X axes are in log10 scale.	168

Figure 7.2 – In each row, the first two subplots are features computed on the Flores101 dev set; the second two subplots are features computed on a subset of our training set. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.	170
Figure 7.3 – Translation results of bilingual experiments with a smaller model (Transformer 6-6). Markers share the same meanings as Figure 7.1. X axes are in log10 scale.	171
Figure 7.4 – Intermediate features and translation results of bilingual experiments with a BPE tokenizer. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.	171
Figure 7.5 – Intermediate features and translation results of bilingual experiments with a 32K vocabulary. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.	172
Figure 7.6 – Intermediate features and translation results of bilingual experiments with byte-fallback. Note that here the UNK rates are all 0, and closeness to the character level can be larger than 1 because one character can be represented by multiple bytes. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.	173
Figure 7.7 – Translation results (spBLEU) of our main multilingual experiments. Marker shapes denote the language pairs (though all pairs share the same NMT model); dash or solid lines represents out-of-English or into-English directions; colors are for each language. E.g., --▲- (en-ta) denotes English to Tamil translation results; -▲ (ta-en) represents Tamil to English translation results. X axes are in log10 scale.	175
Figure 7.8 – Translation results (chrF) of our main multilingual experiments. Markers have the same meanings as Figure 7.7. X axes are in log10 scale.	176
Figure 7.9 – Intermediate features of our main multilingual experiments. Different from Figure 7.7, here, marker shapes and colors both denote the language. E.g., -▲ (ta) denotes Tamil features. X axes are in log10 scale.	176
Figure 7.10 – Intermediate features and translation results of the multilingual experiments with byte-fallback. Markers of the first two subplots have the same meanings as Figure 7.9, and markers of the second two subplots have the same meanings as Figure 7.7. X axes are in log10 scale.	178

CHAPTER 1: INTRODUCTION

Natural language generation (NLG) is distinguished from natural language understanding by its focus on producing natural language output, though eventually, reliable NLG requires a proper understanding of the context and the world. There is a wide range of NLG tasks (Gehrmann et al., 2021), including machine translation (Wu et al., 2016; Bahdanau et al., 2015), text summarization (Nenkova and McKeown, 2012; Liu and Lapata, 2019), data-to-text (Liu et al., 2018; Parikh et al., 2020), story generation (Fan et al., 2018a; Yao et al., 2019), question generation (Heilman and Smith, 2010; Du et al., 2017), etc. Despite the diversity, they are mostly based on the same modeling methodology: autoregressive language modeling, i.e., generating words from left to right. Recently, due to the success of large pretrained language models (LMs) (Radford et al., 2019; Raffel et al., 2020; Lewis et al., 2020a; Zhang et al., 2020a), finetuning pretrained LMs on these diverse tasks has become a *de facto* standard and has improved the performance of diverse tasks significantly. More excitingly, a single large pretrained LM can even perform many tasks in a zero-shot manner simply by giving it different prompts (Brown et al., 2020). Despite this impressive progress, even the strongest model, ChatGPT, still makes a lot of mistakes and is not completely reliable in practice. Moreover, advanced NLG technologies are usually data-hungry and thus only support a few high-resource languages in the world, while leaving a lot of other languages behind. **Therefore, the primary goal of this thesis is to improve the *reliability* and *inclusiveness* of NLG technologies.**

On the one hand, **unreliability** refers to any issue that can make NLG models untrustworthy in practice. These issues are pervasive in the NLG pipeline: data can be noisy, biased, or incorrect (Bommasani and Cardie, 2020); models can produce incoherent (Holtzman et al., 2020), factually inconsistent (Cao et al., 2018; Maynez et al., 2020), or toxic (Gehman et al., 2020) text;

evaluation methods (human or automatic evaluations) are unable to reflect the actual quality of the model (Deutsch and Roth, 2021). All of these problems hinder NLG technologies from being reliably applied in the real world. This thesis focuses on improving NLG reliability by proposing alternative training objectives as well as evaluation methods.

NLG models are usually trained by maximum likelihood estimation (MLE), which does not always align with how the models are evaluated and how we expect the model to behave. Hence, it is necessary to propose **alternative learning objectives beyond MLE**. The first NLG task we try to improve is Question generation (QG). QG is a task to produce a question for an answer span in a text paragraph. One underlying requirement is that the generated question should be answerable by the given answer. However, existing end-to-end QG models, which are trained by MLE, do not have this constraint in their training objectives. And, the output questions are often unanswerable by the answer. In Zhang and Bansal (2019), we propose to use an external pre-trained QA model to verify the answerability of the generated question and use it as a reward to train the QG model via reinforcement learning (RL). We show that our method greatly improved the answerability of questions and achieved state-of-the-art QG performance at the time of publication.

Next, we focus on language modeling. Though human text usually has low perplexity under the model distribution, random sampling from the model often results in incoherent and nonsensical text. Therefore, we believe these models are *over-generalized*, in the sense that the model distribution Q_θ has larger support than the human distribution P . We believe MLE contributes to this over-generalization problem. MLE is equivalent to minimizing the forward cross-entropy (CE) or forward KL divergence, which has a *zero-avoiding* property – avoiding $Q_\theta(x) = 0$ when $P(x) \neq 0$ (Murphy, 2012). Therefore, if there is noise in the data, Q_θ will try to cover the noise as well, which leads the model to *over-generalize*. To address this problem, we propose to **mix MLE with an objective of minimizing the reverse CE** between Q_θ and P ($-\mathbb{E}_{x \sim Q_\theta}[\log P(x)]$), which we call MixCE (Zhang et al., 2023b). Reverse CE reflects human evaluations and can efficiently narrow the over-generalized model distribution down. However, due to the unknown

P , optimizing reverse cross-entropy is intractable. We introduce an approximation of reverse CE which ends up being a simple *self-reinforced* loss function – encouraging the model to generate what it is confident about. We show that compared to MLE, finetuning pretrained LMs with MixCE greatly improved their sampling performance with respect to both automatic and human evaluations.

Evaluation is critical for NLG because not only do we need it to assess and compare different systems but also it can provide useful feedback for model training. How to **design reliable and reproducible evaluation methods** is a long-standing problem. In text summarization, many works conduct *direct human rating*, i.e., humans are asked to rate the summary from certain angles, which suffers from rating subjectivity, difficulty in designing rating criteria, and high expense of time and budget. Another human evaluation protocol, Pyramid or LitePyramid (Nenkova and Passonneau, 2004; Shapira et al., 2019), is proposed to address some of these issues. Given human-written reference summaries, LitePyramid asks humans to break references into *summary content units (SCUs)*; then it asks humans to judge whether each SCU is present in the system summary; finally, the score is the number of present SCUs divided by the total number of SCUs. LitePyramid is easier to implement and more reproducible, but still costly. On the other hand, automatic metrics (e.g., ROUGE (Lin, 2004)) are widely used because they are fast and low-cost, but lots of studies have criticized their unreliability. To find a trade-off between human and automatic evaluations, following LitePyramid, we substitute the human judgment of SCU-presence with a natural language inference (NLI) model (Zhang and Bansal, 2021) which we call Lite²Pyramid. SCU extraction only needs to be conducted once by humans. After having SCUs, Lite²Pyramid can run automatically for evaluating different systems. As long as the same NLI model is being used, the same results will be obtained. Lite²Pyramid greatly outperforms existing automatic metrics. Besides, we also propose to gradually automate SCU extraction using Semantic Role Labeling, resulting in a spectrum of Lite^{2..x}Pyramid metrics and a fully-automatic Lite³Pyramid metric. Overall, we show that combining human and automatic evaluations can help to find a good balance between reproducibility and reliability.

An increasing number of works have been focusing on evaluating the faithfulness of system summaries (Durmus et al., 2020; Wang et al., 2020a) because unfaithfulness, e.g., changing the meaning of the source, is widely spread across text summarization tasks and systems. These works have only focused on abstractive summarization (generating novel sentences) rather than extractive summarization (extracting sentences from the source). However, as we found in (Hu et al., 2022), **extracted summaries can also mislead the audience** by biasing towards one side of the sentiment. Moreover, coreference and discourse issues can also show up across extracted sentences. To systematically study this problem, my recent work (Zhang et al., 2023a) introduced the first error typology with five types of broad unfaithfulness problems that can appear in extractive summaries, including *incorrect coreference*, *incomplete coreference*, *incorrect discourse*, *incomplete discourse*, as well as *other misleading information*. We asked humans to label these problems out of 1600 English summaries produced by 15 diverse extractive systems. We found that 30% of the summaries have at least one of the five issues, which demonstrates that extractive is not faithful. We found that 5 existing faithfulness evaluation metrics for abstraction summarization have poor correlations with human judgment. To remedy this, we proposed a new metric, ExtEval, that is designed for detecting unfaithful extractive summaries and is shown to have the best performance. Overall, we want to remind the community that even though all content is extracted from the source, there is still a chance to be unfaithful. Recently, using our collected data, we test whether large pretrained LMs can score a faithful abstractive summary higher than an unfaithful extractive summary of the same source document (Tam et al., 2023). Unfortunately, we find that LMs almost always prefer the extractive summary despite the fact that it is unfaithful, which paves the ground for more research in this direction.

On the other hand, existing LMs usually only support English or high-resource languages. Among the 6,500 languages spoken or signed in the world today, lots of them are left behind. Supporting as many languages as possible is an important mission of the NLP community. This thesis works towards increasing the **inclusiveness** of NLG research as well as reciprocating the underrepresented language communities via NLG technologies. Throughout my Ph.D. studies, I

have been working on the **language processing of an endangered Native American Language, Cherokee**. In collaboration with Prof. Ben Frey (a linguist, a Cherokee citizen, and a second-language speaker of Cherokee), we collect a Cherokee-English parallel dataset (Zhang et al., 2020b) which is also used in Stanford CS224n NLP course. We develop the first set of Cherokee-English translation systems and an online translation demo (Zhang et al., 2021b). The demo supports both neural and statistical machine translations, provides quality estimation to inform users how trustworthy the translation is, and collects human feedback. The demo has been used by Cherokee speakers and learners and was featured by UNC Research in the headline story during the American Indian Heritage Month in 2021. Besides, we also introduced a more complete roadmap for using NLP to help revitalize endangered languages like Cherokee (Zhang et al., 2022b), in which we proposed suggestions to NLP practitioners, approaches of NLP-assisted language education, and future directions for Cherokee language processing. Eventually, we hope that with the help of NLP technologies, we can increase the number of active speakers of Cherokee and bring it back to day-to-day use.

In addition, to make NLP more inclusive, a lot of effort has been made on developing multilingual models. When multiple languages are involved, usually one single multilingual tokenizer is trained. However, due to the different amounts of data in different languages, **low-resource languages may not be well represented in a multilingual vocabulary**. As a result, they can be excessively tokenized into characters, resulting in long sequences, and some tokens will be considered unknown. Both long sequences and unknown tokens can lead to poor downstream performance. We systematically study how language imbalance in tokenization affects the performance of multilingual translation (Zhang et al., 2022a). We find that translation models are surprisingly robust to language imbalance; nonetheless, better performance is often observed when languages are more balanced.

1.1 Thesis Statement

The goal of this thesis is to make natural language generation (NLG) models more trustworthy and support more diverse languages. Concretely, we introduce alternative NLG training objectives beyond maximum likelihood training, propose more reliable NLG evaluation methods, and extend NLG technologies to support endangered and low-resource languages.

1.2 Overview of Chapters

The remainder of this dissertation is organized into seven chapters. Chapter 2 presents our work on improving the answerability of model-generated questions via reinforcement learning. Chapter 3 presents our work on improving language modeling performance by mixing forward and reverse cross-entropies. Chapter 4 presents our work on combining human and automatic evaluations to achieve a balance between reliability and reproducibility. Chapter 5 presents our work on analyzing broad unfaithfulness problems in extractive summaries. Chapter 6 presents our work on building the Cherokee-English machine translation dataset and system as well as reviewing the roadmap of how NLP can help with language revitalization. Chapter 7 presents our work on studying the language imbalance problem of multilingual tokenizer training. Chapter 8 summarizes the contributions herein and discusses the potential opportunities for future work.

CHAPTER 2: ADDRESSING SEMANTIC DRIFT IN QUESTION GENERATION

2.1 Introduction

Previous QG systems follow an attention-based sequence-to-sequence structure, taking the paragraph-level context and answer as inputs and outputting the question. However, we observed that these QG models often generate questions that semantically drift away from the given context and answer; we call this the “semantic drift” problem. As shown in Figure 2.1, the baseline QG model generates a question that has almost contrary semantics with the ground-truth question, and the generated phrase “the principle of enlightenment” does not make sense given the context. We conjecture that the reason for this “semantic drift” problem is because the QG model is trained via teacher forcing only, without any high-level semantic regularization. Hence, the learned model behaves more like a question language model with some loose context constraint, while it is unaware of the strong requirements that it should be closely grounded by the context and should be answered by the given answer. Therefore, we propose two semantics-enhanced rewards to address this drift: **QPP** and **QAP**. Here, **QPP** refers to **Question Paraphrasing Probability**, which is the probability of the generated question and the ground-truth question being paraphrases; **QAP** refers to **Question Answering Probability**, which is the probability that the generated question can be correctly answered by the given answer. We regularize the generation with these two rewards via reinforcement learning. Experiments show that these two rewards can significantly improve the question generation quality separately or jointly, and achieve the new state-of-the-art performance on the SQuAD QG task.¹

Next, in terms of QG evaluation, previous works have mostly adopted popular automatic evaluation metrics, like BLEU, METEOR, etc. However, we observe that these metrics often fall

¹At the time of publication (mid-2019).

Context: ...during the age of enlightenment, philosophers such as **john locke** advocated the principle in their writings, whereas others, such as thomas hobbes, strongly opposed it. montesquieu was one of the foremost supporters of separating the legislature, the executive, and the judiciary...

Gt: who was an advocate of separation of powers?

Base: who opposed the principle of enlightenment?

Ours: who advocated the principle in the age of enlightenment?

Figure 2.1: An examples of the “semantic drift” issue in Question Generation (“Gt” is short for “ground truth”).

short in properly evaluating the quality of generated questions. First, they are not always correlated to human judgment about answerability (Nema and Khapra, 2018). Second, since multiple questions are valid but only one reference exists in the dataset, these traditional metrics fail to appropriately score question paraphrases and novel generation (shown in Figure 2.3). Therefore, we introduce a QA-based evaluation method that directly measures the QG model’s ability to mimic human annotators in generating QA training data, because ideally, we hope that the QG model can act like a human to ask questions. We compare different QG systems using this evaluation method, which shows that our semantics-reinforced QG model performs best. However, this improvement is relatively minor compared to our improvement on other QG metrics, which indicates improvement on typical QG metrics does not always lead to better question annotation by QG models for generating QA training set.

Further, we investigate how to use our best QG system to enrich QA datasets and perform semi-supervised QA on SQuADv1.1 (Rajpurkar et al., 2016). Following the back-translation strategy that has been shown to be effective in Machine Translation (Sennrich et al., 2016b) and Natural Language Navigation (Fried et al., 2018; Tan et al., 2019), we propose two methods to collect synthetic data. First, since multiple questions can be asked for one answer while there is only one human-labeled ground-truth, we make our QG model generate new questions for existing context-answer pairs in SQuAD training set, so as to enrich it with paraphrased and other novel but valid questions. Second, we use our QG model to label new context-answer pairs from new Wikipedia articles. However, directly mixing synthetic QA pairs with ground-truth data will not lead to improvement. Hence, we introduce two empirically effective strategies: one is a “data

filter” based on QAP (same as the QAP reward) to filter out examples that have low probabilities to be correctly answered; the other is a “mixing mini-batch training” strategy that always regularizes the training signal with the ground-truth data. Experiments show that our method improves both BiDAF (Seo et al., 2017; Clark and Gardner, 2018) and BERT (Devlin et al., 2019) QA baselines by 1.69/1.27 and 1.19/0.56 absolute points on EM/F1, respectively; even without introducing new articles, it can bring 1.51/1.13 and 0.95/0.13 absolute improvement, respectively.

Github repository: <https://github.com/ZhangShiyue/QGforQA>

2.2 Background and Related Work

Question Generation. Early QG studies focused on using rule-based methods to transform statements to questions (Heilman and Smith, 2010; Lindberg et al., 2013; Labutov et al., 2015). Recent works adopted the attention-based sequence-to-sequence neural model (Bahdanau et al., 2015) for QG tasks, taking answer sentence as input and outputting the question (Du et al., 2017; Zhou et al., 2017), which proved to be better than rule-based methods. Since human-labeled questions are often relevant to a longer context, later works leveraged information from the whole paragraph for QG, either by extracting additional information from the paragraph (Du and Cardie, 2018; Song et al., 2018; Liu et al., 2019a) or by directly taking the whole paragraph as input (Zhao et al., 2018; Kim et al., 2018; Sun et al., 2018). A very recent concurrent work applied the large-scale language model pre-training strategy for QG and also achieved a new state-of-the-art performance (Dong et al., 2019). However, the above models were trained with teacher forcing only. To address the exposure bias problem, some works applied reinforcement learning taking evaluation metrics (e.g., BLEU) as rewards (Song et al., 2017; Kumar et al., 2018). Yuan et al. (2017) proposed to use a language model’s perplexity (R_{PPL}) and a QA model’s accuracy (R_{QA}) as two rewards but failed to get significant improvement. Their second reward is similar to our QAP reward except that we use QA probability rather than accuracy as the probability distribution is more smooth. Hosking and Riedel (2019) compared a set of different rewards, including

R_{PPL} and R_{QA} , and claimed none of them improved the quality of generated questions. For QG evaluation, even though some previous works conducted human evaluations, most of them still relied on traditional metrics (e.g., BLEU). However, Nema and Khapra (2018) pointed out the existing metrics do not correlate with human judgment about answerability, so they proposed “Q-metrics” that mixed traditional metrics with an “answerability” score. In our work, we will show QG results on traditional metrics, Q-metrics, as well as human evaluation, and also propose a QA-based QG evaluation.

Question Generation for QA. As the dual task of QA, QG has been often proposed for improving QA. Some works have directly used QG in QA models’ pipeline (Duan et al., 2017; Dong et al., 2017; Lewis and Fan, 2019). Some other works enabled semi-supervised QA with the help of QG. Tang et al. (2017) applied the “dual learning” algorithm (He et al., 2016) to learn QA and QG jointly with unlabeled texts. Yang et al. (2017) and Tang et al. (2018) followed the GAN (Goodfellow et al., 2014) paradigm, taking QG as a generator and QA as a discriminator, to utilize unlabeled data. Sachan and Xing (2018) proposed a self-training cycle between QA and QG. However, these works either reduced the ground-truth data size or simplified the span-prediction QA task to answer sentence selection. Dhingra et al. (2018) collected 3.2M cloze-style QA pairs to pre-train a QA model, then fine-tune with the full ground-truth data which improved a BiDAF-QA baseline. In our paper, we follow the back-translation (Sennrich et al., 2016b) strategy to generate new QA pairs by our best QG model to augment SQuAD training set. Further, we introduce a data filter to remove poorly generated examples and a mixing mini-batch training strategy to more effectively use the synthetic data. Similar methods have also been applied in some very recent concurrent works (Dong et al., 2019; Alberti et al., 2019) on SQuADv2.0. The main difference is that we also propose to generate new questions from existing articles without introducing new articles.

2.3 Question Generation

2.3.1 Base Model

We first introduce our base model which mainly adopts the model architecture from the previous state-of-the-art (Zhao et al., 2018). The differences are that we introduce two linguistic features (POS & NER), apply deep contextualized word vectors, and tie the output projection matrix with the word embedding matrix. Experiments showed that with these additions, our base model results surpass the results reported in Zhao et al. (2018) with significant margins. Our base model architecture is shown in the upper box in Figure 2.2 and described as follow. If we have a paragraph $p = \{x_i\}_{i=1}^M$ and an answer a which is a sub-span of p , the target of the QG task is to generate a question $q = \{y_j\}_{j=1}^N$ that can be answered by a based on the information in p .

Embedding. The model first concatenates four word representations: word vector, answer tag embedding, Part-of-Speech (POS) tag embedding, and Name Entity (NER) tag embedding, i.e., $e_i = [w_i, a_i, p_i, n_i]$. For word vectors, we use the deep contextualized word vectors from ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019). The answer tag follows the BIO² tagging scheme.

Encoder. The output of the embedding layer is then encoded by a two-layer bi-directional LSTM-RNNs, resulting in a list of hidden representations H . At any time step i , the representation h_i is the concatenation of \vec{h}_i and \overleftarrow{h}_i .

$$\begin{aligned}\vec{h}_i &= \overrightarrow{LSTM}([e_i; \vec{h}_{i-1}]) \\ \overleftarrow{h}_i &= \overleftarrow{LSTM}([e_i; \overleftarrow{h}_{i+1}]) \\ H &= [\vec{h}_i, \overleftarrow{h}_i]_{i=1}^M\end{aligned}\tag{2.1}$$

Self-attention. A gated self-attention mechanism (Wang et al., 2017) is applied to H to aggregate the long-term dependency within the paragraph. α_i is an attention vector between h_i and

²“B”, for “Begin”, tags the start token of the answer span; “I”, for “Inside”, tags other tokens in the answer span; “O”, for “Other”, tags other tokens in the paragraph.

each element in H ; u_i is the self-attention context vector for h_i ; h_i is then updated to f_i using u_i ; a soft gate g_i decides how much the update is applied. $\hat{H} = [\hat{h}_i]_{i=1}^M$ is the output of this layer.

$$\begin{aligned}
u_i &= H\alpha_i, \alpha_i = \text{softmax}(H^T W^u h_i) \\
f_i &= \tanh(W^f [h_i; u_i]) \\
g_i &= \text{sigmoid}(W^g [h_i; u_i]) \\
\hat{h}_i &= g_i * f_i + (1 - g_i) * h_i
\end{aligned} \tag{2.2}$$

Decoder. The decoder is another two-layer uni-directional LSTM-RNN. An attention mechanism (Luong et al., 2015) dynamically aggregates \hat{H} at each decoding step to a context vector c_j which is then used to update the decoder state s_j .

$$\begin{aligned}
c_j &= \hat{H}\alpha_j, \alpha_j = \text{softmax}(\hat{H}^T W^a s_j) \\
\tilde{s}_j &= \tanh(W^c [c_j; s_j]) \\
s_{j+1} &= \text{LSTM}([y_j; \tilde{s}_j])
\end{aligned} \tag{2.3}$$

The probability of the target word y_j is computed by a maxout neural network.

$$\begin{aligned}
\tilde{o}_j &= \tanh(W^o [c_j; s_j]) \\
o_j &= [\max\{\tilde{o}_{j,2k-1}, \tilde{o}_{j,2k}\}]_k \\
p(y_j | y_{<j}) &= \text{softmax}(W^e o_j)
\end{aligned} \tag{2.4}$$

In practice, we keep the weight matrix W^e the same as the word embedding matrix and fix it during training. Furthermore, we apply a “pointer network” (Gu et al., 2016) to enable the model to copy words from input.

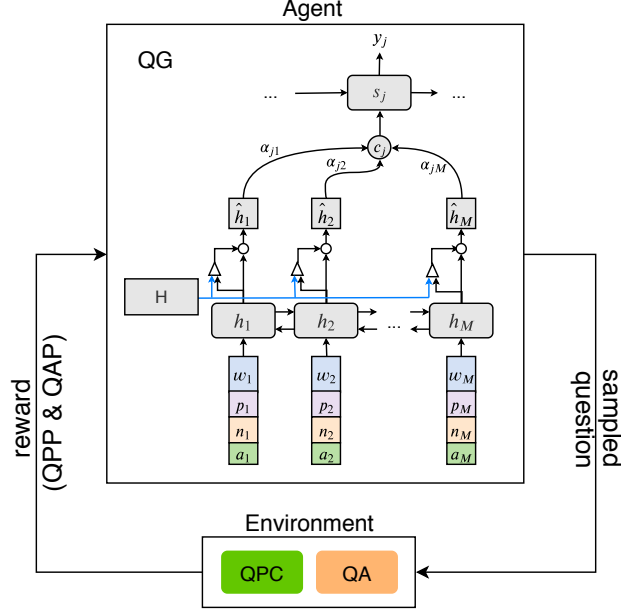


Figure 2.2: The architecture of our semantics-reinforced QG model.

2.3.2 Semantics-Reinforced Model

To address the “semantic drift” problem shown in Figure 2.1, we propose two semantics-enhanced rewards to regularize the generation to focus on generating semantically valid questions.

QPP Reward. To deal with the “exposure bias” problem, many previous works directly used the final evaluation metrics (e.g., BLEU) as rewards to train the generation models (Rennie et al., 2017; Paulus et al., 2018). However, these metrics sometimes fail to evaluate equally to question paraphrases and thus provide inaccurate rewards. Hence, we propose to use a pre-trained question paraphrasing classification (QPC) model to provide paraphrasing probability as a reward. Since paraphrasing is more about semantic similarity than superficial word/phrase matching, it treats question paraphrases more fairly (Example 1 in Figure 2.3). Therefore, we first train a QPC model with Quora Question Pairs dataset. Next, we take it as an environment, and the QG model will interact with it during training to get the probability of the generated question and the ground-truth question being paraphrases as the reward.

Example 1: Fail to score equally to paraphrases	BLEU4	Q-BLEU1	QPP	QAP
Context: ...the university first offered graduate degrees , in the form of a master of arts (ma) , in the the 1854 – 1855 academic year ...				
Gt: in what year was a master of arts course first offered ?				
Gen1: in what year did the university first offer a master of arts ?	37.30	79.39	49.71	34.09
Gen2: when did the university begin offering a master of arts ?	29.58	47.50	46.12	18.18
Example 2: Fail to score appropriately to novel generation				
Context: ...in 1987 , when some students believed that the observer began to show a conservative bias , a liberal newspaper , common sense was published...				
Gt: in what year did the student paper common sense begin publication ?				
Gen1: in what year did common sense begin publication ?	56.29	85.77	92.28	93.94
Gen2: when did the observer begin to show a conservative bias ?	15.03	21.11	13.44	77.15

Figure 2.3: Two examples of where QPP and QAP improve in question quality evaluation.

QAP Reward. Two observations motivate us to introduce QAP reward. First, in a paragraph, usually, there are several facts relating to the answer and can be used to ask questions. Neither the teacher forcing or the QPP reward can favor this kind of novel generation (Example 2 in Figure 2.3). Second, we find semantically-drifted questions are usually unanswerable by the given answer. Therefore, inspired by the dual learning algorithm (He et al., 2016), we propose to take the probability that a pre-trained QA model can correctly answer the generated question as a reward, i.e., $p(a^*|q^s; p)$, where a^* is the ground-truth answer and q^s is a sampled question. Using this reward, the model can not only gets positive rewards for novel generation but also be regularized by the answerability requirement. Note that, this reward is supposed to be carefully used because the QG model can cheat by greedily copying words in/near the answer to the generated question. In this case, even though high QAPs are achieved, the model loses the question generation ability.

Policy Gradient. To apply these two rewards, we use the REINFORCE algorithm (Williams, 1992) to learn a generation policy p_θ defined by the QG model parameters θ . We minimize the loss function $L_{RL} = -E_{q^s \sim p_\theta}[r(q^s)]$, where q^s is a sampled question from the model’s output distribution. Due to the non-differentiable sampling procedure, the gradient is approximated using a

single sample with some variance reduction baseline b :

$$\nabla_{\theta} L_{RL} = -(r(q^s) - b) \nabla_{\theta} \log p_{\theta}(q^s) \quad (2.5)$$

We follow the effective SCST strategy (Rennie et al., 2017) to take the reward of greedy search result q^g as the baseline, i.e., $b = r(q^g)$. However, only using this objective to train QG will result in poor readability, so we follow the mixed loss setting (Paulus et al., 2018): $L_{mixed} = \gamma L_{RL} + (1 - \gamma) L_{ML}$. In practice, we find the mixing ratio γ for QAP reward should be lower, i.e., it needs more regularization from teacher forcing, so that it can avoid the undesirable cheating issue mentioned above. Furthermore, we also apply the multi-reward optimization strategy (Pasunuru and Bansal, 2018) to train the model with two mixed losses alternately with an alternate rate $n : m$, i.e., train with L_{mixed}^{qpp} for n mini-batches, then train with L_{mixed}^{qap} for m mini-batches, repeat until convergence. n and m are two hyper-parameters.

$$\begin{aligned} L_{mixed}^{qpp} &= \gamma^{qpp} L_{RL}^{qpp} + (1 - \gamma^{qpp}) L_{ML} \\ L_{mixed}^{qap} &= \gamma^{qap} L_{RL}^{qap} + (1 - \gamma^{qap}) L_{ML} \end{aligned} \quad (2.6)$$

Experiments show that these two rewards can significantly improve the QG performance separately or jointly, and we achieve new state-of-the-art QG performances, see details in Section 2.6.

2.3.3 QA-Based QG Evaluation

Inspired by the idea that “a perfect QG model can replace humans to ask questions”, we introduce a QA-based evaluation method that measures the quality of a QG model by its ability to mimic human annotators in labeling training data for QA models. The evaluation procedure is described as follows. First, we sample some unlabeled Wikipedia paragraphs with pre-extracted answer spans from HarvestingQA dataset (Du and Cardie, 2018). Second, we make a QG model act as an “annotator” to annotate a question for each answer span. Third, we train a QA model using this synthetic QA dataset. Lastly, we use the QA model’s performance on the original SQuAD

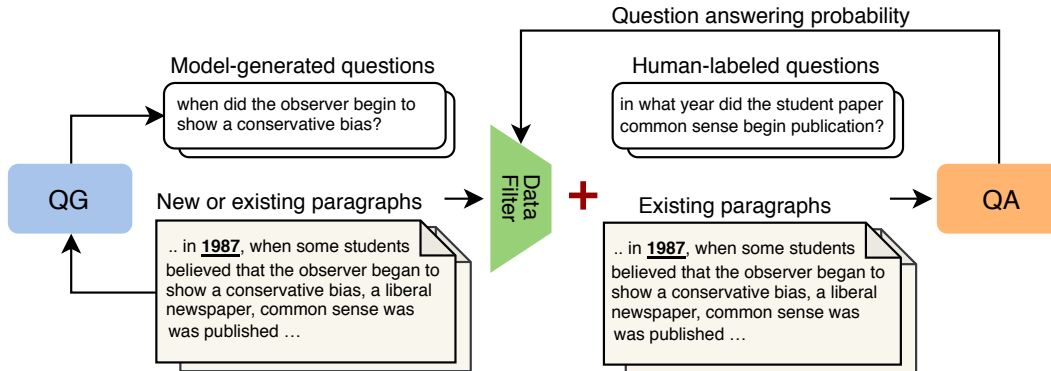


Figure 2.4: Semi-supervised QA: First, a trained QG model generates questions from new or existing paragraphs building up a synthetic QA dataset; Second, a data filter filters out low-QAP synthetic examples and augment the rest to human-labeled QA pairs; Lastly, the QA model is trained with the enlarged QA dataset.

development set as the evaluation for this QG model. The higher this QA performance is, the better the QG model mimics a human’s question-asking ability. We believe that this method provides a new angle to evaluate QG model’s quality and also a more reliable way to choose QG models to conduct data augmentation and semi-supervised QA.

2.4 Semi-Supervised Question Answering

Since one of the major goals of developing QG systems is to generate new QA pairs and augment QA datasets, we investigate how to use our QG system to act as a question annotator, collect new QA pairs, and conduct semi-supervised QA. Figure 2.4 illustrates the overall procedure of our semi-supervised QA approach.

2.4.1 Synthetic Data Generation

To generate synthetic QA pairs, we follow the effective “back translation” approach proposed in Neural Machine Translation (NMT) (Sennrich et al., 2016b). In NMT, the back translation method first obtains synthetic source sentences by running a pre-trained target-to-source translation model on a monolingual dataset of the target language; then, it combines the synthetic and ground-truth translation pairs to train the desired source-to-target translation model. Similarly, in

the QA scenario, the paragraph/answer can be viewed as the “target sentence”, while the question can be taken as the “source sentence”. One tricky difference is that even if the paragraphs can be easily obtained from Wikipedia, there are no answer span labels. Therefore, we use two sources to generate questions from, as discussed below.

Generate from existing articles. In SQuAD (Rajpurkar et al., 2016), each context-answer pair only has one ground-truth question. However, usually, multiple questions can be asked. The diversity lies in question paraphrasing and different facts in the context that can be used to ask the question. Therefore, without introducing new Wikipedia articles, we make our QG model generate diverse questions for the existing context-answer pairs in SQuAD training set by keeping the all beam search outputs for each example.

Generate from new articles. To use unlabeled Wikipedia articles for data augmentation, an automatic answer extractor is indispensable. Some previous works have proposed methods to detect key phrases from a paragraph and automatically extract potential answer spans (Yang et al., 2017; Du and Cardie, 2018; Subramanian et al., 2018). Instead of building up our answer extractor, we directly take advantage of the released HarvestingQA dataset. It contains 1.2M synthetic QA pairs, in which both the answer extractor and the QG model were proposed by Du and Cardie (2018). We use their paragraphs with answer span labels but generate questions with our QG models, and only use their questions for comparison.

2.4.2 Synthetic Data Usage

In practice, we find that directly mixing the synthetic data with the ground-truth data does not improve QA performance. We conjecture the reason is that some poor-quality synthetic examples mislead the learning process of the QA model. Therefore, we propose two empirical strategies to better utilize synthetic data.

QAP data filter. In the literature of *self-training*, similar issues have been discussed that using model-labeled examples to train the model will amplify the model’s error. Later works proposed

co-training or tri-training that uses two or three models as judges and only keeps examples that all models agree on (Blum and Mitchell, 1998; Zhou and Li, 2005). Sachan and Xing (2018) also designed question selection oracles based on curriculum learning strategy in their QA-QG self-training circle. In this paper, we simply design a data filter based on our QAP measure (same definition as the QAP reward) to filter poor-quality examples. We think if one question-answer pair has a low QAP, i.e., the probability of the answer given the question is low, it is likely to be a mismatched pair. Hence, we filter synthetic examples with $QAP < \epsilon$, where ϵ is a hyper-parameter that we will tune for different synthetic datasets.

Mixing mini-batch training. When conducting semi-supervised learning, we do not want gradients from ground-truth data are overwhelmed by synthetic data. Previous works (Fried et al., 2018; Dhingra et al., 2018) proposed to first pre-train the model with synthetic data and then fine-tune it with ground-truth data. However, we find when the synthetic data size is small (e.g., similar size as the ground-truth data), catastrophic forgetting will happen during fine-tuning, leading to similar results as using ground-truth data only. Thus, we propose a “mixing mini-batch” training strategy, where for each mini-batch we combine half mini-batch ground-truth data with half mini-batch synthetic data, which keeps the data mixing ratio to 1:1 regardless of what the true data size ratio is. In this way, we can have the training process generalizable to different amounts of synthetic data and keep the gradients to be regularized by ground-truth data.

2.5 Experiment Setup

2.5.1 Datasets

QG. For QG, we use the SQuAD-based QG dataset³ first introduced by Du et al. (2017) which was the most widely-used QG dataset in previous works (Song et al., 2018; Zhao et al., 2018; Du and Cardie, 2018; Kim et al., 2018; Sun et al., 2018). It was derived from SQuADv1.1 (Rajpurkar et al., 2016). Since the testing set is not open, they sampled 10% articles from the train-

³<https://github.com/xinyadu/nqg/tree/master/data>

ing set as the testing set, and the original development set is still used for validation. For the QA-based QG evaluation, we obtain new paragraphs with pre-extracted answer spans from HarvestingQA (Du and Cardie, 2018). Without using their provided questions, we have different QG models act as “annotators” to generate questions, and then use the different QG-labeled synthetic datasets to train QA models. We use the same dev-test setup as described below.

QA. For QA, we use SQuADv1.1 (Rajpurkar et al., 2016). Previous semi-supervised QA works sampled 10% from training set as the testing set (Yang et al., 2017; Dhingra et al., 2018). Since we want to use the full training set in semi-supervised QA setup without any data size reduction, we instead split the original development set in half for validation and testing respectively. For semi-supervised QA, first, without introducing new articles, we generate new questions for SQuAD training set by keeping all beam search outputs. Second, with introducing new articles, we obtain new paragraphs with pre-extracted answer spans from HarvestingQA (Du and Cardie, 2018). Without using their provided questions, we use our best QG model to label questions. Meanwhile, we investigate the influence of synthetic data size, so we sample 10% to 100% examples from HarvestingQA, which are denoted as H1-H10 in our experiments.

2.5.2 Evaluation Metrics

QG. First, we use three traditional automatic evaluation metrics: BLEU4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE-L (Lin, 2004). Second, we adopt the new “Q-metrics” proposed by Nema and Khapra (2018), and we only use “Q-BLEU1” that was shown to have the highest correlation with human judgments on SQuAD. We also take the QPP and QAP rewards as two additional evaluation metrics. Further, we conduct a pairwise human comparison between our baseline and best QG models. The detailed human evaluation setup is described in the following. For the QA-based QG evaluation, we use the same QA evaluation metrics as follows.

QA. Following the standard evaluation method for SQuADv1.1 (Rajpurkar et al., 2016), we use Exact Match (EM) and F1 as two metrics.

Human Evaluation. We performed pairwise human evaluation between our baseline and the QPP&QAP multi-reward model on Amazon Mechanical Turk. We selected human annotators that are located in the US, have an approval rate greater than 98%, and have at least 10,000 approved HITs. We showed the annotators an input paragraph with the answer bold in the paragraph and two questions generated by two QG models (randomly shuffled to anonymize model identities). We then asked them to decide which one is better or choose “non-distinguishable” if they are equally good/bad. We give human three instructions about what is a good question: first, “answerability” – a good question should be answerable by the given answer; “making sense” – a good question should be making sense given the surrounding context; “overall quality” – a good question should be as fluent, non-ambiguous, semantically compact as possible. Ground-truth questions were not provided to avoid simple matching with ground-truth.

2.5.3 Implementation Details

QG. For ELMo-QG, we first tokenize and obtain the POS/NER tags by Stanford CoreNlp toolkit⁴, then lower-case the entire dataset. We use 2-layer LSTM-RNNs for both encoder and decoder with hidden size 600. Dropout with a probability of 0.3 is applied to the input of each LSTM-RNN layer. We use the pre-trained character-level word embedding from ELMo (Peters et al., 2018) both as our word embedding and output-projection matrix, and keep it fixed. We use Adam (Kingma and Ba, 2015) as optimizer with learning rate 0.001 for teacher forcing and 0.00001 for reinforcement learning. Batch size is set to 32. For stability, we first pre-train the model with teacher forcing until convergence, then fine-tune it with the mixed loss. Hyper-parameters are tuned on development set: $\gamma^{qpp} = 0.99$, $\gamma^{qap} = 0.97$, and $n : m = 3 : 1$. We use beam search with beam size 10 for decoding and apply a bi-gram/tri-gram repetition penalty as proposed in Paulus et al. (2018). For BERT-QG, we simply replace the ELMo used above to BERT (Devlin et al., 2019). To match with BERT’s tokenization, we use the WordPiece tokenizer to tokenize each word obtained above and extend the POS/NER tags to each word piece. Decoder’s word-

⁴<https://stanfordnlp.github.io/CoreNLP/>

piece outputs will be mapped to normal words by post-processing. Hyper-parameters are tuned on development set: $\gamma^{qpp} = 0.99$, $\gamma^{qap} = 0.97$, and $n : m = 1 : 3$.

QA. For BiDAF-QA, we implement the BiDAF+Self-attention architecture proposed by Clark and Gardner (2018). We use GRUs for all RNN layers with hidden size 90 for GRUs and 180 for linear layers. Dropout with a probability of 0.2 is applied to the input of each GRU-RNN layer. We optimize the model using Adadelata with batch size 64. We also add ELMo to both the input and output of the contextual GRU-RNN layer as proposed in (Peters et al., 2018). To match with QG model’s setup, we also apply lower-case on QA datasets. For BERT-QA, we use the pre-trained uncased BERT-base model⁵ and fine-tune it on QA datasets.

QPC. For ELMo-QPC, we follow the model architecture proposed by Conneau et al. (2017). First, two input questions are embedded with ELMo (Peters et al., 2018). Second, the embedded questions are encoded by two 2-layer bidirectional LSTM-RNNs separately with hidden size 512. Next, a max-pooling layer outputs the sentence embedding of each question, denoted by q_1 and q_2 . Lastly, we input $[q_1, q_2, |q_1 - q_2|, q_1 * q_2]$ to an MLP to predict whether these two questions are paraphrases or not. This QPC model is trained using the Quora Question Pairs⁶ dataset. We use Adam (Kingma and Ba, 2015) as optimizer with learning rate 0.0004 and batch size 64. This model obtained 86% accuracy on QQP development set. For BERT-QPC, we also use the pre-trained uncased BERT-base model and fine-tune it on QQP dataset, which obtained 90% accuracy on QQP development set.

⁵<https://github.com/google-research/bert>

⁶<https://tinyurl.com/y2y8u5ed>

⁷They actually used the reversed dev-test setup as opposed to the original setup used in Du et al. (2017) and Du and Cardie (2018) (see Section 3.1 in Zhao et al. (2018)). Thus, we also conducted the reversed dev-test setup and our QPP&QAP model yields BLEU4/METEOR/ROUGE-L=20.76/24.20/48.91.

	BLEU4	METEOR	ROUGE-L	Q-BLEU1	QPP	QAP
Du and Cardie (2018)	15.16	19.12	–	–	–	–
Zhao et al. (2018) ⁷	16.38	20.25	44.48	–	–	–
Our baseline (w. ELMo)	17.00	21.44	45.89	47.80	27.29	45.15
+ BLEU4	17.72	22.13	46.52	49.07	27.09	45.96
+ METEOR	17.84	22.41	46.18	49.09	26.70	46.52
+ ROUGE-L	17.78	22.28	46.51	49.23	27.06	46.31
+ QPP	18.25	22.62	46.45	49.59	28.13	47.63
+ QAP	18.12	22.52	46.45	49.27	27.49	48.76
+ QPP&QAP	18.37	22.65	46.68	49.63	28.03	48.37

Table 2.1: The performance of different QG models.

QPP&QAP	Our baseline	Tie
160	131	9

Table 2.2: Pairwise human evaluation between our baseline and QPP&QAP multi-reward model.

2.6 Results

2.6.1 Question Generation

Baselines. First, as shown in Table 2.1, our baseline QG model obtains a non-trivial improvement over previous best QG system (Zhao et al., 2018) which proves the effectiveness of our newly introduced setups: introduce POS/NER features, use deep contextualized word vectors (from ELMo or BERT), and tie output projection matrix with non-trainable word embedding matrix. Second, we apply three evaluation metrics as rewards to deal with the exposure bias issue and improve performance. All the metrics are significantly⁸ ($p < 0.001$) improved except QPP, which supports that high traditional evaluation metrics do not always correlate to high semantic similarity.

Semantics-reinforced models. As shown in Table 2.1, when using QAP and QPP separately, all metrics are significantly ($p < 0.001$) improved over our baseline and all metrics except ROUGE-L are significantly ($p < 0.05$) improved over the models using traditional metrics

⁸The significance tests in this paper are conducted following the bootstrap test setup (Efron and Tibshirani, 1994).

Data	Du and Cardie	Our baseline	QPP & QAP
H1	53.20/65.47	55.06/67.83	55.89/68.26
H2	53.40/66.28	56.23/ 69.23	56.69 /69.19
H3	53.12/65.57	57.14 /69.39	57.05/ 70.17
S+H1	71.16/80.75	71.94/81.26	72.20/81.44
S+H2	72.02/81.00	72.03/81.38	72.22/81.81
S+H3	71.48/81.02	72.61/81.46	72.69/82.22

Table 2.3: The QA-based evaluation results for different QG systems. The two numbers of each item in this table are the EM/F1 scores. All results are the performance on our QA test set. “S” is short for “SQuAD”.

as rewards. After applying multi-reward optimization, our model performs consistently best on BLEU4/METEOR/ROUGE-L and Q-BLEU1. Notably, using one of these two rewards will also improve the other one at the same time, but using both of them achieves a good balance between these two rewards without exploiting either of them and results in the consistently best performance on other metrics, which is a new state-of-the-art result.

Human evaluation. Table 2.2 shows the MTurk anonymous human evaluation study, where we do a pairwise comparison between our baseline and QPP&QAP model. We collected 300 responses in total, 160 of which voted the QPP&QAP model’s generation is better, 131 of which favors the baseline model, and 9 of which selected non-distinguishable.

QA-based evaluation. As shown in Table 2.3, we compare three QG systems using QA-based evaluation on three different amounts of synthetic data and their corresponding semi-supervised QA setups (without filter). It can be observed that both our baseline and our best QG model can significantly improve the synthetic data’s QA performance, which means they can act as better “annotators” than the QG model proposed by Du and Cardie (2018). However, our best QG model only has a minor improvement over our baseline model, which means significant improvement over QG metrics does not guarantee significant better question annotation ability.

⁹“Data Size” counts the total number of examples in training set (after filter). In Table 2.6, “New Data Size” only counts # examples generated from articles outside SQuAD.

	Filter	Data Size ⁹	EM	F1
H1 only	$\epsilon = 0.0$	120k	54.55	67.91
	$\epsilon = 0.2$	84k	61.18	71.65
	$\epsilon = 0.4$	69k	61.97	72.48
	$\epsilon = 0.6$	55k	60.38	70.51
	$\epsilon = 0.8$	40k	57.47	66.48
SQuAD+H1	$\epsilon = 0.0$	207k	72.97	82.18
	$\epsilon = 0.2$	171k	73.88	82.72
	$\epsilon = 0.4$	156k	73.47	82.62
	$\epsilon = 0.6$	142k	73.96	82.81
	$\epsilon = 0.8$	127k	73.65	82.77

Table 2.4: The effect of QAP-based synthetic data filter. We filter out the synthetic data with $QAP < \epsilon$. All results are the performance on our QA development set.

	Data	Data Size	EM	F1
Dev set	SQuAD	87k	72.52	81.79
	+ Beam5	399k	74.33	83.19
	+ Beam10	706k	74.44	83.23
	+ Beam15	853k	74.25	82.75
	+ DivBeam10	595k	74.44	83.00
Dev set	+ H1	142k	73.96	82.81
	+ H2	255k	74.19	82.84
	+ H4	424k	74.42	82.82
	+ H6	506k	74.27	82.97
	+ H8	705k	74.64	83.14
	+ H10	930k	74.27	82.97
Test set	SQuAD	87k	71.92	81.26
	+ Beam10	706k	73.43	82.39
	+ H8	705k	73.61	82.53
	+ Beam10 + H8	1.3M	73.43	82.11

Table 2.5: The results of our semi-supervised QA method using a BiDAF-QA model.

2.6.2 Semi-Supervised Question Answering

Effect of the data filter. As shown in Table 2.4, when using synthetic data only, adding the data filter can significantly improve QA performance. In terms of semi-supervised QA, the improvement is relatively smaller, due to the regularization from ground-truth data, but still consistent and stable.

Methods	New Data Size	EM	F1
Dhingra et al. base	0	71.54	80.69
+Cloze	3.2M	71.86	80.80
Our base	0	72.19	81.52
+Beam10	0	73.93	82.81
+H8	705k	74.12	82.83

Table 2.6: The comparison with the previous semi-supervised QA method. All results are the performance on the full development set of SQuAD, i.e., our QA test + development set.

Semi-supervised QA. Table 2.5 demonstrates the semi-supervised QA results. Without introducing new articles, we keep beam search outputs as additional questions. It can be seen that using beam search with beam size 10 (+Beam10) improves the BiDAF-QA baseline by 1.51/1.13 absolute points on the testing set. With introducing new articles, the best performance (+H8) improves the BiDAF-QA baseline by 1.69/1.27 absolute points on the testing set. We also combine the two best settings (Beam10+H8), but it does not perform better than using them separately. We conduct two ablation studies on the development set. First, we compare beam search with different beam sizes and diverse beam search (Li et al., 2016), but all of them perform similarly. Second, increasing the size of synthetic data from H1 to H10, the performance saturates around H2-H4. We also observed that when using a big synthetic dataset, e.g., H10, the model converges even before all examples were used for training. Based on these results, we conjecture that there is an upper bound of the effect of synthetic data which might be limited by the QG quality. To further improve the performance, more diverse and tricky questions need to be generated. To show how QG models help or limit the QA performance, we include some synthetic QA examples in Section 2.6.4. Finally, we compare our semi-supervised QA methods with Dhingra et al. (2018). As shown in Table 2.6, with no or less new data injection, our methods achieve larger improvements over a stronger baseline than their method.

	BLEU4	METEOR	ROUGE-L	Q-BLEU1	QPP	QAP	QA-Eval (H1)
Du and Cardie (2018)	15.16	19.12	–	–	–	–	55.11/66.40
Our baseline (w. BERT)	18.05	22.41	46.57	49.38	29.08	54.61	58.63/69.97
+ QPP	18.51	22.87	46.65	49.97	30.14	55.67	60.49/71.81
+ QAP	18.65	22.91	46.76	50.09	30.09	57.51	60.12/71.14
+ QPP & QAP	18.58	22.87	46.76	50.01	30.10	56.39	59.11/70.87

Table 2.7: The performance of our stronger BERT-QG models.

	Data	Data Size	EM	F1
Dev set	SQuAD	87k	81.88	88.80
	+ Beam10	668k	82.34	88.97
	+ H10	664k	82.88	89.53
Test set	SQuAD	87k	80.25	88.23
	+ Beam10	668k	81.20	88.36
	+ H10	664k	81.03	88.79
	+ Beam10 + H10	1.2M	81.44	88.72

Table 2.8: The results of our semi-supervised QA method using a stronger BERT-QA model.

2.6.3 QG and QA Results with BERT

The Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) has recently improved a lot of NLP tasks by substantial margins. To verify if our improvements still hold on BERT-based baselines, we propose a BERT-QG baseline and test our two semantics-enhanced rewards; further, we conduct our semi-supervised QA method on a BERT-QA baseline.

BERT-QG. Without modifying our QG model’s architecture, we simply replaced ELMo used above with BERT. Table 2.7 shows that our BERT-QG baseline improves previous ELMo-QG baseline by a large margin; meanwhile, our QPP/QAP rewards significantly improve the stronger QG baseline and achieve the new state-of-the-art QG performance w.r.t both traditional metrics and QA-based evaluation. One difference is that the QAP-only model has the overall best performance instead of the multi-reward model. Note that, we also obtain the QPP and QAP rewards from BERT-based QPC and QA models, respectively.

BERT-QA. Using our QAP-reinforced BERT-QG model, we apply the same semi-supervised QA method on a BERT-QA baseline. As shown in Table 2.8, though with smaller margins, our

method improves the strong BERT-QA baseline by 1.19/0.56 absolute points on testing set; even without introducing new articles, it obtains 0.95/0.13 absolute gains.

2.6.4 Examples

Figure 2.5 shows some synthetic QA examples generated by our QG models. On SQuAD, the first two examples show our QG models generate some paraphrases or novel questions that enrich the dataset; the last two examples show our QG models generate easier or wrong questions that limit the semi-supervised QA’s performance. On HarvestingQA, our QG models can output better questions than Du and Cardie (2018) did but still generate some wrong questions.

2.7 Conclusion

We proposed two semantics-enhanced rewards to regularize a QG model to generate semantically valid questions, and introduced a QA-based evaluation method that directly evaluates a QG model’s ability to mimic human annotators in generating QA training data. Experiments showed that our QG model achieves new state-of-the-art performances. Further, we investigated how to use our QG system to augment QA datasets and conduct semi-supervised QA via two synthetic data generation methods along with a data filter and mixing mini-batch training. Experiments showed that our approach improves both BiDAF and BERT QA baselines even without introducing new articles.

Examples generated on SQuAD
Context: ...new york city consists of five boroughs, each of which is a separate county of new york state...
Ground-truth: how many boroughs does new york city contain ?
ELMo-QG: how many boroughs make up new york city ?
BERT-QG: new york city consists of how many boroughs ?
Context: ...gendün gyatso traveled in exile looking for allies. however, it was not until 1518 that the secular phagmodru ruler captured lhasa from the rinbung, and thereafter the gelug was given rights to conduct the new years prayer...
Ground-truth: when was gelug was given the right to conduct the new years prayer ?
ELMo-QG: in what year did the secular phagmodru ruler take over lhasa ?
BERT-QG: when did the secular phagmodru ruler capture lhasa from the rinbung ?
Context: ...chopin attended the lower rhenish music festival in aix-la-chapelle with hiller, and it was there that chopin met felix mendelssohn. after the festival, the three visited düsseldorf... they spent what mendelssohn described as “a very agreeable day”, playing and discussing music at his piano...
Ground-truth: what two activities did frédéric do while visiting for a day in düsseldorf with mendelssohn and hiller ?
ELMo-QG: what did mendelssohn do at his piano ?
BERT-QG: what did chopin do at his piano ?
Context: ...to limit protests, officials pushed parents to sign a document, which forbade them from holding protests, in exchange of money, but some who refused to sign were threatened ...
Ground-truth: what has happened to some who refuse to agree to not protest ?
ELMo-QG: what did some who refused to sign ?
BERT-QG: what did the officials refused to sign ?
Examples generated on HarvestingQA
Context: ...nigeria prior to independence was faced with sectarian tensions and violence... some of the ethnic groups like the ogoni, have experienced severe environmental degradation due to petroleum extraction ...
Du and Cardie (2018): what is the main reason for the ethnic groups ?
ELMo-QG: why has nigeria experienced severe environmental degradation ?
BERT-QG: why have the ogoni experienced severe environmental degradation ?
Context: ...vietnam is located on the eastern indochina peninsula... at its narrowest point in the central quang binh province , the country is as little as across...
Du and Cardie (2018): where is the country ’s country located ?
ELMo-QG: in what province is vietnam located ?
BERT-QG: what province is vietnam ’s narrowest point ?
Context: ...the ottoman islamic legal system was set up differently from traditional european courts ...
Du and Cardie (2018): where was the ottoman islamic legal system set ?
ELMo-QG: the ottoman islamic legal system was set up from what ?
BERT-QG: what was the ottoman islamic legal system set up differently from ?
Context: ...the eastern shore of virginia is the site of wallops flight facility , a rocket testing center owned by nasa...
Du and Cardie (2018): what is the eastern shore of virginia owned by ?
ELMo-QG: what facility is owned by nasa ?
BERT-QG: what is the name of the rocket facility located by nasa ?

Figure 2.5: Some synthetic QA examples generated by our QG models.

CHAPTER 3: TRAINING LANGUAGE MODELS BY MIXING FORWARD AND REVERSE CROSS-ENTROPIES

3.1 Introduction

Rapid advances in pre-trained large-scale autoregressive language models (LMs) have dramatically improved the performance of a variety of tasks (Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022c; Chowdhery et al., 2022). However, these systems still struggle in many open-ended generation settings, where they are asked to produce long text following a short prompt. In these cases, we seek systems that generate sensible, coherent, fluent, and engaging, or in short, *human-like* text (Pillutla et al., 2022).

Different decoding strategies to generate such text from pretrained LMs suffer from different degeneration problems. Unbiased sampling¹ usually results in incoherent and nonsensical text, while greedy and beam searches often get stuck in repetition loops (Holtzman et al., 2020). These observations suggest that the learned LM distribution Q_θ still varies substantially from the human LM distribution P . A possible reason is that the autoregressive modeling of Q_θ gives a non-zero probability to every possible sequence of tokens, while many sequences are impossible under P . Nevertheless, we still hope that $Q_\theta(x)$ is as small as possible when $P(x) = 0$. To this end, maximum likelihood estimation (MLE), i.e., minimizing the cross-entropy (CE) $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, is the most widely used objective to train $Q_\theta(x)$ using sequences sampled from P . In an idealized setting, with unlimited training data and model capacity, as well as a perfect optimizer, fitting Q_θ with MLE will learn a distribution as close to P as we like. However, in practice, we only have finite and noisy data.

¹Unbiased sampling is vanilla random sampling, i.e., sampling with temperature=1.0. It is also called ancestral sampling (Eikema and Aziz, 2020) or pure sampling (Holtzman et al., 2020). We call it unbiased sampling because it allows unbiased exploration of the model distribution.

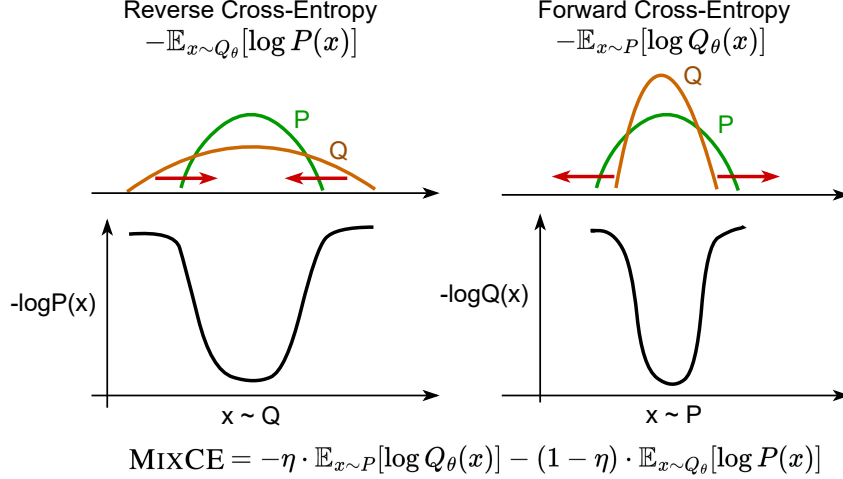


Figure 3.1: MixCE combines two complementary driving forces: reverse CE helps narrow the model distribution Q_θ down when it is broader than data distribution P , while forward CE helps broaden Q_θ out when it is narrower than P . Note that $\log P(x)$ is infinite when $P(x) = 0$. But in practice, we use $\log P(x) = \sum_t \log(P(x_t|x_{<t}) + \epsilon)$ to avoid $\log 0$ and $\epsilon = 1e - 30$.

We argue that the MLE objective only weakly penalizes generations x from Q_θ that are “bad”, in the sense that $P(x) = 0$. When Q_θ puts a small amount of probability mass onto $P(x) = 0$ space, MLE cannot sufficiently discourage this behavior (see Figure 3.2). Moreover, minimizing forward CE, $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, is equivalent to minimizing the forward KL divergence between P and Q_θ , i.e., $\text{KL}(P||Q_\theta) = \mathbb{E}_{x \sim P}[\log P(x)/Q_\theta(x)]$. Forward KL has a *zero-avoiding* property – avoiding $Q_\theta(x) = 0$ when $P(x) \neq 0$ (Murphy, 2012). Therefore, if there is noise in the data, Q_θ will try to cover the noise as well, which leads the model to *over generalize*, in the sense of putting non-trivial probability mass over $P(x) = 0$ generations (Huszár, 2015; Theis et al., 2016; Ott et al., 2018; Kang and Hashimoto, 2020). As a result, we observe samples from the model deviating from human-like text. A common strategy is to modify the decoding method, e.g., top- k , top- p , typical, contrastive (Fan et al., 2018a; Holtzman et al., 2020; Meister et al., 2022; Li et al., 2022) samplings, to tailor the model distribution Q_θ in a post-hoc manner to avoid unwanted generations. In contrast, our approach differs: how can we obtain a better Q_θ to obviate the need for these sampling strategies?

We propose a novel training objective for autoregressive LMs – MixCE that **Mixes** the forward and reverse **Cross-Entropies**: $-\eta \cdot \mathbb{E}_{x \sim P}[\log Q_\theta(x)] - (1 - \eta) \cdot \mathbb{E}_{x \sim Q_\theta}[\log P(x)]$. MixCE

can be understood in two ways (see Section 3.3.1). First, we want model generations to be high-quality as well as diverse. Reverse cross-entropy reflects how we conduct human evaluations, sampling from the model Q_θ and evaluating it by the human P , where the focus is text *quality*. Forward cross-entropy emphasizes the *diversity* of model generations (Hashimoto et al., 2019). Second, MixCE works similarly to a mixture of the forward and reverse KL divergences. The reverse KL divergence ($\text{KL}(Q_\theta||P)$) is *zero-forcing* – forcing $Q_\theta(x) = 0$ when $P(x) = 0$ – and thus more strongly penalizes generating non-human-like samples compared to MLE. Overall, MixCE combines two complementary driving forces to better fit Q_θ to P (Figure 3.1).

Unfortunately, optimizing reverse cross-entropy is intractable because we do not know P . Hence, we propose an approximation of the reverse cross-entropy (see Section 3.3.2), which ends up being a *self-reinforced* loss function that encourages the model to produce generations in which it is already confident. This loss function has the same computational complexity as forward cross-entropy, making MixCE easy to implement and as fast as MLE.

We demonstrate the effectiveness of MixCE in both a synthetic setting, where the “human” distribution P is known, as well as a real setting. For the synthetic case, we evaluate six learning objectives: MixCE, MixCE* (MixCE without approximation), forward KL (=MLE), reverse KL, the mixture of two KL divergences, and Jensen–Shannon (JS) divergence. We show that MixCE* works slightly worse than the mixture of KLs while outperforming other objectives, and MixCE works worse than MixCE* but generally outperforms MLE. In real settings, we finetune GPT-2 (Radford et al., 2019) of different sizes on three English text domains using MixCE or MLE. Our results show that, compared to MLE, unbiased sampling from MixCE-finetuned models produces text with diversity (Meister et al., 2022) closer to the human text, higher Coherence (Su et al., 2022), higher Mauve (Pillutla et al., 2021), and preferred by humans. When using top- p sampling (Holtzman et al., 2020) and carefully tuning p , generations from MLE-finetuned models are similar to those generated from MixCE-finetuned models. Nonetheless, MixCE models have tuned p values closer to 1, implying a less noisy model distribution. In addition, we modify

the original Mauve to make it more robust to spurious features (e.g., text length), under which MixCE still improves over MLE when using unbiased sampling.

Github repository: <https://github.com/bloomberg/MixCE-acl2023>

3.2 Background and Related Work

3.2.1 Autoregressive Language Modeling

Language generation is mostly based on the autoregressive language modeling methodology. The generation of one word is conditioned on previously generated words, $Q_\theta(x_t|x_{<t})$, and the final probability of the sequence x is the product of probabilities of each step, $Q_\theta(x) = \prod_t Q_\theta(x_t|x_{<t})$. Early works build n-gram neural LMs (Bengio et al., 2000) and then RNN-based LMs (Mikolov et al., 2010), and now Transformer (Vaswani et al., 2017) has become the dominant architecture. Language generation models have either a decoder-only (Mikolov et al., 2010) or an encoder-decoder architecture (Sutskever et al., 2014; Bahdanau et al., 2015). In this work, we focus on decoder-only LMs. In recent years, many large-scale pre-trained decoder-only LMs have been introduced (Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022c; Chowdhery et al., 2022). They can be finetuned for downstream tasks and even perform surprisingly well in a zero-shot or few-shot manner. Despite the impressive performance, language *degeneration* is one of the key issues that remain to be solved.

3.2.2 Language *Degeneration*

According to Holtzman et al. (2020), language degeneration refers to output text that is *bland*, *incoherent*, or *gets stuck in repetitive loops*. It is widely observed in open-ended generations from pretrained LMs. Two commonly observed patterns of degeneration are the incoherent text from unbiased sampling and the repetitive text from greedy or beam search. Degeneration also appears in sequence-to-sequence generation tasks but in a slightly different form (Stahlberg and Byrne, 2019).

There is no agreement on what causes degeneration. Ott et al. (2018) attribute it to data noise and the smooth class of model functions. It is inherent in the model’s structure to have support everywhere, in particular, because all probabilities are produced by softmax, which is strictly positive. Therefore, Hewitt et al. (2022) assume that an LM distribution is the true data distribution plus a uniform-like smoothing distribution. Based on the observation that human-like text has a large but not too large likelihood under the learned LM distribution (Zhang et al., 2021a), a lot of works propose empirically useful decoding methods beyond unbiased sampling and greedy/beam search (Fan et al., 2018a; Holtzman et al., 2020; Eikema and Aziz, 2020; Basu et al., 2021; Meister et al., 2022; Li et al., 2022; Hewitt et al., 2022; Su et al., 2022; Krishna et al., 2022). One of these approaches is the canonical top- p (or nucleus) sampling method (Holtzman et al., 2020), which samples from top tokens that take up p proportion (e.g., 95%) of the probability mass at each decoding step. Even though these decoding methods work impressively well, they are post-hoc fixes rather than learning the LM accurately in the first place. Therefore, some other works criticize the MLE training objective and propose alternative loss functions.

3.2.3 Objectives Beyond MLE

Unlikelihood training (Welleck et al., 2020; Li et al., 2020) was proposed to penalize repetition (or any undesirable phenomenon) explicitly during training. The idea is to minimize the likelihood of a set of negative tokens at each generation step during training. The selection of negative tokens is pre-defined, e.g., tokens that appear often in the previous context. MixCE shares the same goal with unlikelihood training – matching human distribution, but provides a more general approach without targeting any specific problem.

Similar to our motivation, Kang and Hashimoto (2020) think that the zero-avoiding property of MLE makes the model sensitive to dataset noise. To cover these noisy examples, the model has to put non-trivial probability mass on the $P(x) = 0$ area. To combat this problem, they propose a loss truncation method that drops high-loss (low-likelihood) examples during training time.

Pang and He (2021) want to address the mismatch of learning objective and human evaluation (likelihood vs. quality) and introduce the GOLD algorithm to approximate reverse cross-entropy. Our approximation is similar to theirs but has a different derivation process (see Section 3.3.2). Moreover, GOLD is evaluated on controlled generation tasks (e.g., summarization, translation) in which the goal is to generate one high-quality text for each input, and diversity is not so important. In contrast, if we train the LM only with reverse CE till convergence, the model will deterministically produce the most likely text for each prompt, which is undesirable for an LM. Therefore, mixing forward and reverse CEs is necessary.

The idea of MixCE is also relevant to GANs (Goodfellow et al., 2014). GANs optimize the Jensen–Shannon (JS) divergence between model and data distributions. Essentially, JS divergence is also for balancing the two driving forces of forward and reverse KL divergences (Huszár, 2015), and it has been successfully used for evaluating LM-generated text (Pillutla et al., 2021). However, probably due to the discrete nature of text, GANs have not been well applied to LM training. Caccia et al. (2020) show that previous language GANs often trade off diversity for quality.

A more relevant past work to ours is Popov and Kudinov (2018) which finetunes LMs with the sum of the forward cross-entropy loss and reverse KL divergence. They train a discriminator like what GAN does to estimate reverse KL. Differently, we directly approximate reverse cross-entropy without training an additional discriminator.

Concurrently, with the same motivation as ours, Ji et al. (2023) propose to replace MLE with minimizing the total variation distance (TVD) (Van Handel, 2014) between data and model distributions. Notably, their final approximation of TVD, which they call TaiLr, is equivalent to forward cross-entropy when the hyperparameter $\gamma = 0$ and equals our approximated reverse cross-entropy when $\gamma = 1$.

3.3 Methodology

3.3.1 MixCE

Our MixCE learning objective for training LMs is the combination of forward and reverse cross-entropies, written as

$$-\eta \cdot \mathbb{E}_{x \sim P}[\log Q_\theta(x)] - (1 - \eta) \cdot \mathbb{E}_{x \sim Q_\theta}[\log P(x)] \quad (3.1)$$

where η is the mixing ratio. When $\eta = 1$, it becomes the normal MLE objective; and when $\eta = 0$, it is the reverse cross-entropy only.

The MixCE loss can be understood in two ways. First, reverse and forward cross-entropy (CE) emphasize *quality* and *diversity* respectively. The reverse CE, $-\mathbb{E}_{x \sim Q_\theta}[\log P(x)]$, focuses on *quality* because it resembles how we conduct human evaluations – sampling from the model Q_θ and evaluating it by the human P . In human evaluations, the focus is more on the quality of the model-generated text. So, it is possible that a model always generates the same few high-quality texts, but still gets high human evaluation scores. This is similar to the *mode collapse* problem of GANs. The forward CE, $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, instead focuses more on *diversity* because it needs any sample from P to have a non-trivial probability under Q_θ (Hashimoto et al., 2019). Note that it does not mean forward CE has zero effect on quality, rather, the model likelihood $Q_\theta(x)$ only loosely correlates with the human-perceived quality of x (Zhang et al., 2021a).

Second, we hypothesize that MixCE works similarly to a mixture of forward and reverse KL divergences, which we will show empirically in our synthetic experiments (Section 3.4.1). On the one hand, minimizing forward KL is equivalent to optimizing forward CE. On the other hand, reverse KL divergence, $\mathbb{E}_{x \sim Q_\theta}[\log \frac{Q_\theta(x)}{P(x)}]$, has two parts: reverse CE and negative entropy of Q_θ , $\mathbb{E}_{x \sim Q_\theta}[\log Q_\theta(x)]$. Reverse CE is minimized when the model deterministically outputs the most likely example, i.e., $Q_\theta(x) = \delta(\text{the most likely } x \text{ under } P)$. Instead, minimizing the negative entropy (maximizing the entropy) of the model encourages it to be as uncertain as possible, i.e.,

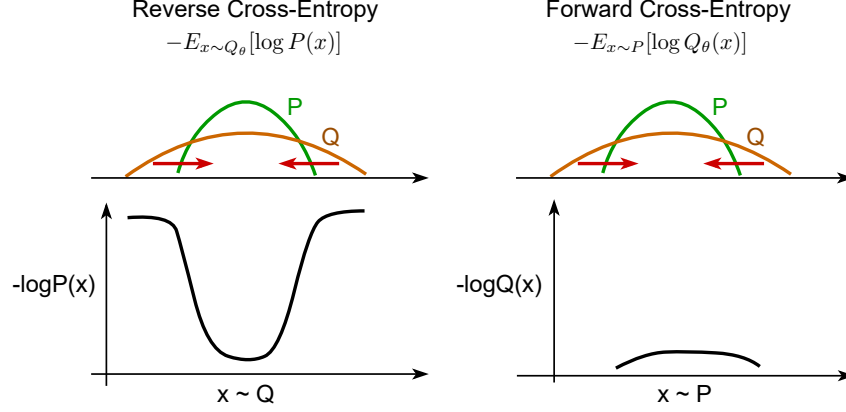


Figure 3.2: Forward CE only weakly penalizes the model Q_θ when it puts a small amount of probability mass onto $P(x)=0$ space. And the loss magnitude is much smaller than what we will get from reverse CE.

having a large support and uniform distribution. This entropy term counteracts the narrowing-down effect of reverse CE. As discussed above, forward CE pushes the Q distribution to fully cover the support of P . In this case, forward CE can also help counteract the narrowing-down effect of reverse CE, i.e., the maximizing entropy term becomes less important when forward CE is present. Hence, we think it is reasonable to drop it from reverse KL.

Overall, MixCE combines two complementary training signals, as shown in Figure 3.1. Reverse CE prevents the model distribution from being broader than the data distribution, while forward CE is more helpful for preventing the model distribution from being narrower than the data distribution. Although forward CE also has non-zero loss when the model distribution is too wide, its loss magnitude is much smaller than what reverse CE provides, as shown in Figure 3.2. When data is clean, two CEs work jointly to help learn the data distribution better. When data is noisy, the mixing ratio η allows us to trade-off between emphasizing a good coverage of the data and putting more weight on the actually high-quality sequences.

3.3.2 Optimization of Reverse CE

Optimizing MixCE is non-trivial. The obstacle is to minimize the reverse CE, $-\mathbb{E}_{x \sim Q_\theta} [\log P(x)]$ with respect to θ . To this end, we need to know P and to have a differentiable sampling opera-

tion from Q_θ . In our synthetic experiments (Section 3.4.1), we use a distribution P of our own construction and use Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to make the sampling operation differentiable.

However, in a real setting, we do not know P . To deal with this, we take the following steps to derive an approximated reverse cross-entropy (we omit the negative sign for simplicity):

$$\nabla_\theta \mathbb{E}_{x \sim Q_\theta} [\log P(x)] \quad (3.2)$$

$$\approx \nabla_\theta \mathbb{E}_{x \sim Q_\theta} [P(x)] \quad (3.3)$$

$$= \sum_x \nabla_\theta Q_\theta(x) P(x) \quad (3.4)$$

$$= \sum_x Q_\theta(x) \nabla_\theta \log Q_\theta(x) P(x) \quad (3.5)$$

$$= \sum_x P(x) Q_\theta(x) \nabla_\theta \log Q_\theta(x) \quad (3.6)$$

$$= \mathbb{E}_{x \sim P} [Q_\theta(x) \nabla_\theta \log Q_\theta(x)] \quad (3.7)$$

$$= \mathbb{E}_{x \sim P} \left[\prod_{t=1}^T Q_\theta(x_t | x_{<t}) \sum_{t=1}^T \nabla_\theta \log Q_\theta(x_t | x_{<t}) \right] \quad (3.8)$$

$$\approx \mathbb{E}_{x \sim P} \left[\sum_{t=1}^T Q_\theta(x_t | x_{<t}) \nabla_\theta \log Q_\theta(x_t | x_{<t}) \right] \quad (3.9)$$

First, from (3.2) to (3.3), we substitute expected log-likelihood by *expected accuracy*. Irsoy (2019) shows that expected accuracy is a comparable or better alternative loss function to cross-entropy for classification tasks. Then, following the Policy Gradient theorem (Williams, 1992; Sutton et al., 1999), we get (3.4) and (3.5), where we view model Q_θ as the policy and $P(x)$ as the reward we want to optimize for the whole sequence. Next, we switch from the expectation of Q_θ to the expectation of P (from (3.5) to (3.6) and (3.7)), so that we can use the offline samples from P (data samples in the training set) instead of online sampling from Q_θ . We unfold $Q_\theta(x)$, which results in (3.8). Until this point, theoretically, we are already able to optimize the model using Equation (3.8) without knowing P . However, the product of $Q_\theta(x_t | x_{<t})$ has a very

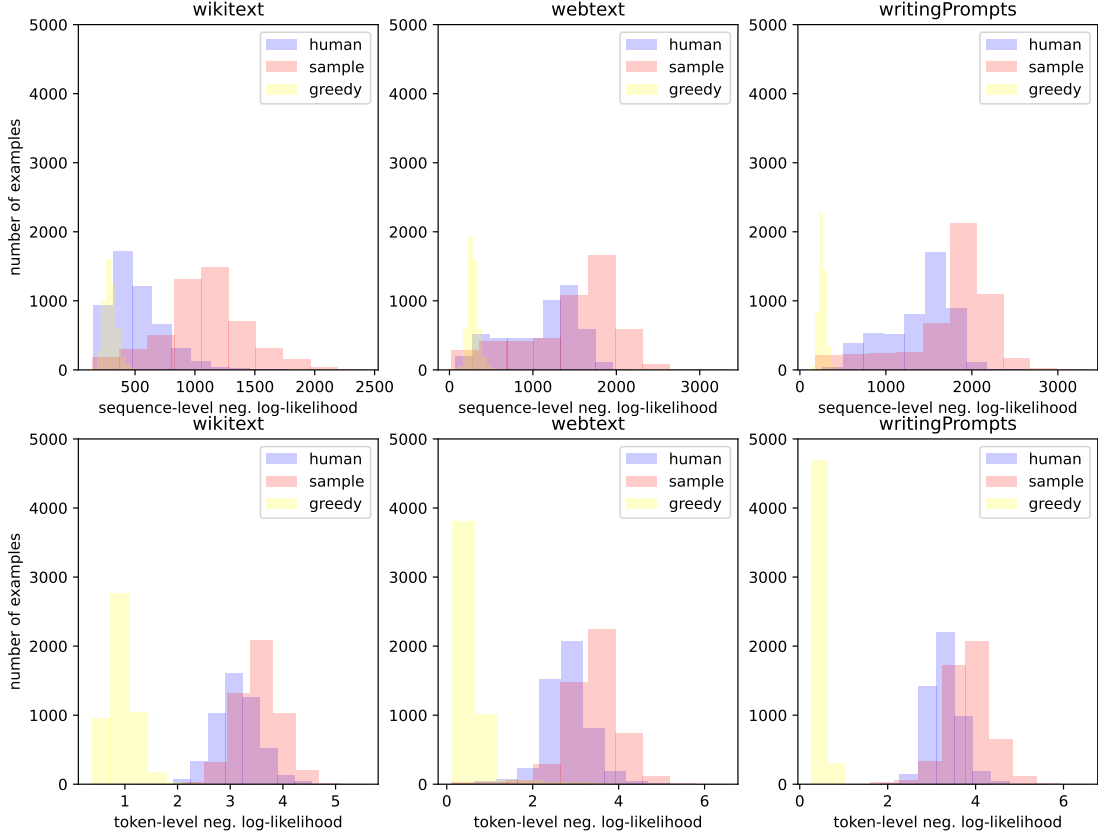


Figure 3.3: The histograms of sequence-level and token-level negative log-likelihoods of human texts and model generations from GPT-2 large.

high variance, and in practice, it underflows when T is large. Therefore, we apply a final rough approximation that leads to (3.9).

Equations (3.8) and (3.9) are apparently not equivalent to each other. Nonetheless, they have similar effects. Intuitively, in (3.8), we weigh the gradients of each sequence differently based on their sequence-level probabilities, $Q_\theta(x)$; in other words, it promotes high-likelihood sequences. Similarly, (3.9) weighs gradients at each step by $Q_\theta(x_t|x_{<t})$, i.e., promoting high-likelihood tokens at each step. So essentially, they both *encourage the model to produce generations in which it is already confident*. We call it a *self-reinforced* objective.

To further illustrate why this *self-reinforced* objective (Equation (3.8) or (3.9)) makes sense and their shortcomings, we conduct an analysis using GPT-2 large (Radford et al., 2019). We first sample 5000 pieces of text from WikiText, WebText, and WritingPrompts, respectively, and

we call them *human* texts. Then, using the first 50 tokens of each human text as a prompt, we get 5000 sampling and greedy search generations from pretrained GPT-2 large (max generation length = 512). Next, we use the same model to score human texts and model generations and get the sequence-level and token-level negative log-likelihoods. Figure 3.3 shows the histograms of these negative log-likelihoods.

In Figure 3.3, we take the human text histogram (in blue) as a proxy of *human distribution* and the sampling text histogram (in red) as a proxy of *model distribution*. As you can see, the support of model distribution usually contains the support of human distribution. It supports our previous claim that MLE-trained models tend to over-generalize. Meanwhile, at both the sequence and the token levels, the model on average assigns a higher probability to human text than to text sampled from the model. Therefore, when we promote high-probability sequences or tokens, it is equivalent to pushing the model distribution toward the human distribution. However, we need to avoid overly pushing it to the extremely high-probability region where greedy search outputs locate (in yellow) because they are known to be poor-quality and repetitive. Also, as shown in the figure, when promoting high-probability *sequences*, even if we overdo it, it will still be within the support of human distribution. In contrast, when promoting high-probability *tokens*, it can go outside the support of the human distribution, which is the drawback of Equation (3.9) compared to Equation (3.8). Lastly, if we train the model only with the self-reinforced objective till convergence, it is inevitable to end up with a model that can only output greedy search generations. Hence, we need to combine it with the forward cross-entropy.

Finally, combining forward CE and Equation (3.9), our approximated MixCE objective is to maximize

$$\mathbb{E}_{x \sim P} \left[\sum_{t=1}^T (\eta + (1 - \eta) \cdot Q_{\theta}(x_t | x_{<t})) \nabla_{\theta} \log Q_{\theta}(x_t | x_{<t}) \right], \quad (3.10)$$

which has the same computational complexity as MLE. Since $Q_{\theta}(x_t | x_{<t})$ (which is around 0.017 to 0.13 when using GPT-2) is strictly lower than 1, the gradient from approximated reverse CE (Equation (3.9)) is smaller than that from forward CE. Therefore, it is important to tune η to balance the effects of two CEs.

3.3.3 Connection to Pang and He (2021)

Similarly, Pang and He (2021) also propose to approximate reverse CE, and the resulting GOLD algorithm is similar to our Equation 3.9. Here, we would like to clarify the difference and connection.

The following equation is the start policy gradient equation used by Pang and He (2021).

$$\mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_t \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

They used different notations from ours. π_θ is the same as our Q_θ , i.e., $\pi_\theta(a_t | s_t)$ is the same as our $Q_\theta(x_t | x_{<t})$. \hat{Q} is the accumulated future reward from timestamp t , $\sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, γ is the decay factor and $r_{t'}$ is the reward for each step. We will discuss \hat{Q} in detail later.

Then, they apply importance sampling to sample from a different behavioral policy π_b . Since they also use examples from the training set, their π_b is the same as our human (or data) distribution P .

$$\mathbb{E}_{\tau \sim \pi_b} \left[\sum_t w_t \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

w_t is the importance weight. They use a per-action approximation: $w_t \approx \frac{\pi_\theta(a_t | s_t)}{\pi_b(a_t | s_t)}$, which is similar to how we get Equation 3.9 from Equation 3.8.

Since π_b is unknown, they assume a uniform distribution: $\pi_b \approx 1/N$ (N is the number of training examples). Hence, their final approximated gradient is:

$$\mathbb{E}_{\tau \sim \pi_b} \left[\sum_t \pi_\theta(a_t | s_t) \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

They define $r_{t'}$ and \hat{Q} in three ways. The first is called δ -reward, i.e., $\hat{Q} = 1$. In this case, their final gradient is exactly the same as our Equation 3.9. However, as you can see, we take a different path of derivation. Instead of using this δ -reward, our \hat{Q} is the sequence-level reward $P(x)$. The reward $P(x)$ nicely helps us to switch from the expectation of Q_θ to the expectation of P (from

Equation 3.5 to Equation 3.7). Therefore, without assuming a uniform distribution of π_b , our π_b is just P .

When using the other two rewards, they also need to know P . To address this, they use an MLE-pretrained model as a proxy of P .

Overall, we introduce a different derivation approach for approximating reverse CE. Moreover, as we mentioned in Section 3.2.3, Pang and He (2021) focused on improving controlled generation tasks where the focus is on the quality of the text, while we focus on open-ended generations where quality and diversity are both important. Therefore, we mix reverse CE with forward CE to form our MixCE learning objective.

3.4 Experiments

3.4.1 Synthetic Experiments

We first conduct experiments in a synthetic ideal setting, where we know P , to show the effectiveness of mixing two cross-entropies with or without approximation. Moreover, during evaluation, we can directly compare the learned model parameters against the ground truth parameters of P .

Define the “human” LM P . We start by defining P as a bi-gram LM. Bi-gram means that the prediction of the next token only depends on the immediately previous token, i.e., $P(x_t|x_{t-1})$. Therefore, P is determined by a transition matrix among words $\mathbf{M} \in \mathbb{R}^{V \times V}$ (V =vocabulary size) and a start token probability distribution $\boldsymbol{\pi} \in \mathbb{R}^V$, i.e., stochastic finite-state automata. The last token in the vocabulary is the end-of-sequence (EOS) token. For simplicity, we initialize $\boldsymbol{\pi}$ as a uniform distribution. To initialize \mathbf{M} , we use two methods. The first is **random initialization**. We sample categorical distributions from a Dirichlet ($\alpha=0.5$) prior to initialize each row of \mathbf{M} . However, one remaining problem is that P has support everywhere. To have $P = 0$ areas, we randomly assign 0s to a certain percent of values in each row of \mathbf{M} and then re-normalize to sum

to 1.² We test 3 percentages: 10%, 50%, and 90%. The second is **initialization using real data**. We sample 5000 pieces of text from WebText (Radford et al., 2019), count the occurrence of bi-grams, and then use the occurrence to initialize \mathbf{M} . In this case, there are naturally 0s in \mathbf{M} , and the larger the vocabulary size is, the sparser \mathbf{M} is. No matter which initialization is used, we reserve the last row of \mathbf{M} for EOS and it has all 0s, i.e., will not transit to any token. We set the vocabulary size $V=20, 50, 100, 500$, or 1000 .³

Learn an LM Q_θ . We implement model Q_θ as a simple neural bigram LM. Given the word embedding e_{i-1} of the previous token x_{i-1} , the next token is predicted via a neural network f :

$$h_{i-1} = \text{Dropout}(\text{ReLU}(\mathbf{W}_1 e_{i-1} + \mathbf{b}_1)),$$

$$Q(x_i|x_{i-1}) = \text{Softmax}(\mathbf{W}_2 h_{i-1} + \mathbf{b}_2),$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ (d is the hidden dimension size), $\mathbf{b}_1 \in \mathbb{R}^d$, $\mathbf{W}_2 \in \mathbb{R}^{d \times V}$, and $\mathbf{b}_2 \in \mathbb{R}^V$ are model parameters. After training this model, the learned transition matrix can be obtained by $\mathbf{M}' = f(\mathbf{E})$, \mathbf{E} is the word embedding matrix.

Synthetic data. We sample sequences from P . We set the max sequence length as 500. We sample 50K and 5K sequences as the training and validation set, respectively. There is no test set because we directly compare the learned transition matrix \mathbf{M}' to the gold \mathbf{M} during evaluation.

Metrics. (1) **avg. js**: we compute the JS divergence between each row (except the last row) of \mathbf{M}' and the corresponding row in \mathbf{M} , and then average across rows. This metric evaluates the overall divergence of \mathbf{M}' from \mathbf{M} , and equals 0 iff $\mathbf{M}' = \mathbf{M}$; (2) **avg. 0s**: we get the probabilities from \mathbf{M}' from positions where the corresponding gold probabilities are 0 in \mathbf{M} , and take their average. If $\mathbf{M}' = \mathbf{M}$, avg. 0s = 0, but vice versa is not true.

²When we assign 0s, we make sure every token has non-zero transition probability to EOS.

³Our defined bi-gram LMs are always *tight*, i.e., do not “leak” probability mass onto infinite sequences because we make sure that all accessible tokens also have non-zero paths to other tokens. Please refer to Du et al. (2022) for the proof.

Vocab	Objective	Random (10%)		Random (50%)		Random (90%)		WebText	
		avg. js	avg. 0s	avg. js	avg. 0s	avg. js	avg. 0s	avg. js	avg. 0s
	Gold	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	For. KL	3.65e-4	1.80e-4	7.40e-4	1.44e-4	7.56e-4	9.10e-5	9.93e-4	1.79e-4
	Rev. KL	3.41e-3	5.56e-6	1.36e-1	7.42e-6	1.87e-1	1.54e-6	3.93e-3	1.95e-6
	Mix KLS	3.11e-4	7.11e-5	4.89e-4	5.15e-5	4.01e-4	2.67e-5	9.91e-4	1.11e-5
	JS	5.68e-3	1.17e-5	2.14e-1	4.88e-5	2.14e-1	5.24e-4	1.12e-2	5.84e-6
	MixCE*	4.92e-4	1.59e-4	8.12e-4	1.05e-4	4.87e-4	2.95e-5	1.36e-3	1.19e-4
	MixCE	3.31e-4	1.57e-4	7.02e-4	1.25e-4	7.08e-4	8.49e-5	1.00e-3	1.79e-4
50	For. KL	6.01e-3	1.21e-3	6.47e-3	5.65e-4	2.18e-3	8.90e-5	4.30e-3	4.77e-4
	Rev. KL	2.03e-2	2.01e-5	4.29e-1	1.53e-3	4.11e-1	4.55e-6	3.48e-2	5.30e-5
	Mix KLS	4.65e-3	1.29e-4	4.45e-3	2.80e-4	1.54e-3	3.41e-5	3.91e-3	2.83e-4
	JS	1.03e-1	9.03e-5	4.74e-1	1.40e-3	4.24e-1	1.25e-5	9.23e-3	2.48e-5
	MixCE*	5.20e-3	6.84e-4	4.49e-3	3.72e-4	1.48e-3	2.70e-5	3.94e-3	2.75e-4
	MixCE	5.96e-3	1.20e-3	6.47e-3	5.64e-4	2.03e-3	7.70e-5	4.29e-3	4.77e-4
100	For. KL	3.34e-2	2.49e-3	3.56e-2	1.44e-3	6.98e-3	1.49e-4	9.70e-3	3.10e-4
	Rev. KL	2.30e-1	1.79e-3	5.57e-1	3.62e-4	5.30e-1	6.25e-6	1.00e-1	4.04e-5
	Mix KLS	2.98e-2	4.66e-4	2.74e-2	2.10e-4	5.04e-3	6.34e-5	9.19e-3	1.84e-4
	JS	2.38e-1	1.06e-3	5.53e-1	9.69e-4	5.18e-1	1.32e-3	1.73e-1	5.56e-4
	MixCE*	3.10e-2	1.73e-3	2.85e-2	9.16e-4	5.12e-3	6.00e-5	9.61e-3	1.87e-4
	MixCE	3.29e-2	2.44e-3	3.56e-2	1.41e-3	7.01e-3	1.50e-5	9.69e-3	3.16e-6
500	For. KL	1.56e-1	1.57e-3	2.39e-1	1.49e-3	1.93e-1	8.45e-4	4.60e-2	1.78e-4
	Rev. KL	2.94e-1	9.91e-4	6.78e-1	2.76e-6	6.49e-1	2.33e-6	3.05e-1	1.68e-5
	Mix KLS	1.55e-1	1.45e-3	2.32e-1	8.60e-4	1.70e-1	6.83e-4	4.27e-2	1.33e-4
	JS	2.95e-1	9.78e-4	5.34e-1	7.19e-4	5.75e-1	1.35e-3	2.78e-1	3.84e-5
	MixCE*	1.55e-1	1.45e-3	2.34e-1	1.38e-3	1.69e-1	6.71e-4	4.23e-2	1.29e-4
	MixCE	1.55e-1	1.56e-3	2.35e-1	1.46e-3	1.88e-1	6.28e-4	4.53e-2	1.64e-4
1000	For. KL	1.83e-1	8.95e-4	2.93e-1	8.80e-4	3.65e-1	7.31e-4	8.10e-2	1.50e-4
	Rev. KL	2.86e-1	6.12e-4	6.85e-1	1.21e-6	6.68e-1	3.88e-6	3.30e-1	6.26e-6
	Mix KLS	1.80e-1	8.64e-4	2.91e-1	8.57e-4	3.50e-1	6.86e-4	7.50e-2	1.17e-4
	JS	2.88e-1	6.11e-4	4.59e-1	5.97e-4	5.80e-1	7.73e-4	3.02e-1	1.93e-5
	MixCE*	1.83e-1	8.64e-4	2.92e-1	8.58e-4	3.50e-1	6.84e-4	7.44e-2	1.14e-4
	MixCE	1.83e-1	8.92e-4	2.92e-1	8.76e-4	3.48e-1	6.71e-4	7.94e-2	1.42e-4

Table 3.1: Synthetic experimental results. Random (10%, 50%, 90%) randomly initializes \mathbf{M} and sets 10% or 50% or 90% of the probabilities to 0. WebText means initializing \mathbf{M} by the bi-gram occurrence in the WebText data. Gold refers to the results when $\mathbf{M}'=\mathbf{M}$. *avg. js* is our main metric, which represents the average JS divergence between \mathbf{M} and \mathbf{M}' (please see the definition of *avg. 0s* in text). Each number is a 5-seed average, and Table 3.2 shows the 95% confidence intervals of some experiments.

Vocab	Objective	WebText	
		avg. js	avg. 0s
1000	For. KL	$8.10\text{e-}2 \pm 2.45\text{e-}4$	$1.50\text{e-}4 \pm 5.58\text{e-}7$
	MixCE*	$7.44\text{e-}2 \pm 2.46\text{e-}4$	$1.14\text{e-}4 \pm 6.15\text{e-}7$
	MixCE	$7.94\text{e-}2 \pm 2.15\text{e-}4$	$1.42\text{e-}4 \pm 5.05\text{e-}7$

Table 3.2: Synthetic experimental results with 95% confidence intervals. WebText means initializing \mathbf{M} by the bigram occurrence in the WebText data.

Objectives. (1) **Forward KL**, $\text{KL}(P||Q_\theta) = \mathbb{E}_{x \sim P}[\log P(x)/Q_\theta(x)]$, which is equivalent to MLE; (2) **Reverse KL**, $\text{KL}(Q_\theta||P) = \mathbb{E}_{x \sim Q_\theta(x)}[\log Q_\theta(x)/P(x)]$; (3) **Mixture of two KLs**, $\eta \cdot \text{KL}(P||Q_\theta) + (1 - \eta) \cdot \text{KL}(Q_\theta||P)$; (4) **JS**, we use a general definition of JS divergence (Huszár, 2015), $\eta \cdot \text{KL}(P||M) + (1 - \eta) \cdot \text{KL}(Q_\theta||M)$, where $M = \eta \cdot P + (1 - \eta) \cdot Q_\theta$; ⁴ (5) **Oracle mixture of cross-entropies** (MixCE*), where we use the known P . (6) **Approximated mixture of cross-entropies** (MixCE), where we assume P is unknown. Except for Forward KL and MixCE, the other four objectives all need to sample from Q_θ and require gradients to pass through this sampling operation. To this end, we use Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to make sampling differentiable.

Model selection. During training, we check the validation loss (the value of the objective function) after every epoch and only save the best checkpoint that has the lowest validation loss. For objectives with η , we choose the best η based on the avg. js result on the validation set. We report a 5-seed average for each experiment. The search space of η is [0.99, 0.9, 0.5, 0.1, 0.01]. Selected best η s are reported in Table 3.10.

Results. Table 3.1 shows the results of our synthetic experiments. Across 4 kinds of initialization of \mathbf{M} and 5 vocabulary sizes, we observe some common patterns. First, the mixture of two KLs often gets the best avg. js compared to other objectives, and MixCE* usually comes second. This supports our expectation that the mixture of two cross-entropies approximates the mixture of two KLs (Section 3.3.1), as well as demonstrates that combining two KLs or CEs can help learn the data distribution more accurately compared to MLE. Second, the approximated MixCE

⁴When $\eta = 0.5$, it is the same as the objective of GAN (Goodfellow et al., 2014). But instead of using GAN’s min-max loss, we directly optimize JS because we know P .

usually under-performs MixCE* but outperforms forward KL (MLE). Third, reverse KL generally works best for the avg. 0s metric, due to its property of *zero-forcing* – forcing $Q_\theta(x) = 0$ when $P(x) = 0$. Lastly, JS divergence oftentimes works similarly to reverse KL, which is consistent with the observation made by Caccia et al. (2020) – language GANs trade off diversity for quality. As the magnitudes of both avg. js and avg. 0s are fairly small, we examine the 95% confidence intervals under one synthetic experimental setting – initializing the transition matrix \mathbf{M} by the bigram occurrence in the WebText data and setting vocabulary size as 1000. Table 3.2 contains the results. We can see that 95% confidence intervals are small enough to maintain the trend of the results.

3.4.2 GPT-2 Experiments

Next, we test MixCE in a real setting where we do not know P , but we have finite samples from P . We use GPT-2 (Radford et al., 2019) as the LM Q_θ . Though GPT-2 models are already pre-trained by MLE, for simplicity, we use different objectives to finetune it. We test GPT-2 in 3 sizes: small (24M), medium (355M), and large (774M). See more implementation details in Section 3.5.3.

Real data. We use English text data from 3 domains: (1) WikiText (Merity et al., 2016): text from Wikipedia; (2) WebText (Radford et al., 2019): text from the Web. It was used for pretraining GPT-2; and (3) WritingPrompts (Fan et al., 2018a): text from the writing prompts forum of Reddit. We sample from each of these 3 datasets to form our training, development, and test sets. By default, our training/development/test set contains 50K/5K/5K examples. Please find more details about these datasets in Section 3.5.3.

Metrics. (1) **Perplexity (ppl)** is defined as $e^{-\frac{1}{N \cdot T} \sum_N \sum_T \log_e Q_\theta(x_t | x_{<t})}$, where N is the number of examples and T is the sequence length. Perplexity is not necessarily correlated with human perceived quality (Zhang et al., 2021a). (2) **Diversity (div)**: following Meister et al. (2022), we define n -gram diversity as the average fraction of unique vs. total n -grams for $n \in \{1, 2, 3, 4\}$ in each piece of text. (3) **Mauve** (Pillutla et al., 2021) compares model-generated text against

human text via a KL divergence curve and is the state-of-the-art metric for open-ended text generation. We use Mauve as our primary metric for model selection. (4) **Coherence (coh)** (Su et al., 2022) computes the cosine similarity between the embedding of prompt and the embedding of continuation, and embeddings are from SimCSE (Gao et al., 2021). All metrics are *the closer to human scores the better*.

Objectives. Since we have no access to P , we can only implement two out of the six objectives we test in the synthetic setting: (1) **MLE**, which is equal to forward CE or forward KL; (2) **MixCE**, the approximated mixture of cross-entropies.

Decoding. We use **unbiased sampling** (see footnote 1) as our primary decoding method as it allows us to explore the learned distribution in an unbiased way (Eikema and Aziz, 2020). Additionally, we test **top- p sampling** (Holtzman et al., 2020) to check if MixCE is complementary to advanced decoding methods, and we carefully tune p based on the Mauve score on the development set. For each text, we take the first 50 tokens (by GPT-2 tokenizer) as the prompt and set the max generation length as 512.

Model selection. We finetune the model for 5 epochs on the training set and save the best checkpoint with the lowest dev loss. We select the best mixing ratio η and the best p based on the Mauve score on the dev set. The search space of η is [0.99, 0.9, 0.7, 0.5, 0.3, 0.1, 0.01, 0.0] and that of p is [0.85, 0.87, 0.89, 0.91, 0.93, 0.95, 0.97, 0.99]. Selected best η s are reported in Table 3.11. Best p s are reported in Table 3.4. Metric scores are reported on the test set and are 3-run averages because sampling is stochastic.

Results. Table 3.3 shows unbiased sampling results of models in different sizes and finetuned with different objectives on three datasets. As you can see, MixCE-finetuned models usually get worse perplexity but consistently better diversity, mauve, and coherence, compared to MLE-finetuned models. Table 3.4 shows top- p sampling results from the same models as Table 3.3. Since perplexity will not change as the decoding method changes, we instead report the selected best p in this table. It can be seen that after carefully applying top- p sampling, MixCE-finetuned

Model Size	Objective	WikiText				WebText				WritingPrompts			
		ppl	div	mauve	coh	ppl	div	mauve	coh	ppl	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
Small	MLE	26.98	0.91	0.67	0.556	21.45	0.87	0.90	0.555	28.45	0.87	0.85	0.397
	MixCE	35.04	0.87	0.93	0.567	21.69	0.85	0.92	0.565	28.79	0.86	0.89	0.403
Medium	MLE	20.43	0.90	0.73	0.573	15.92	0.87	0.88	0.560	22.72	0.88	0.89	0.414
	MixCE	25.92	0.88	0.95	0.584	16.51	0.83	0.93	0.585	23.04	0.86	0.91	0.419
Large	MLE	18.24	0.90	0.75	0.567	14.13	0.87	0.81	0.570	21.95	0.87	0.87	0.425
	MixCE	23.44	0.88	0.95	0.578	14.66	0.82	0.94	0.592	21.04	0.86	0.94	0.429

Table 3.3: Unbiased sampling results of models finetuned by MLE or MixCE on three datasets. For all metrics, the closer to the human scores the better. **Bold** numbers are the ones that are closer to human scores in each setting. Each number is a 3-run average.

Model Size	Objective	WikiText				WebText				WritingPrompts			
		best p	div	mauve	coh	best p	div	mauve	coh	best p	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
Small	MLE	0.85	0.89	0.93	0.584	0.93	0.84	0.94	0.580	0.97	0.86	0.90	0.410
	MixCE	0.99	0.87	0.95	0.568	0.99	0.84	0.93	0.571	0.99	0.85	0.90	0.407
Medium	MLE	0.85	0.88	0.95	0.602	0.93	0.85	0.95	0.592	0.97	0.86	0.92	0.428
	MixCE	0.99	0.87	0.96	0.590	0.99	0.81	0.93	0.594	0.99	0.85	0.92	0.427
Large	MLE	0.87	0.89	0.96	0.594	0.95	0.84	0.87	0.593	0.99	0.86	0.89	0.430
	MixCE	0.99	0.87	0.97	0.580	0.99	0.81	0.94	0.601	0.99	0.86	0.94	0.435

Table 3.4: Top- p sampling results of the same models as Table 3.3. Since changing the decoding method will not affect perplexity, we report the selected best p instead.

models work on par with MLE-finetuned models for diversity, mauve, and coherence. Nonetheless, the best p for MixCE models is always 0.99, while MLE models have smaller and more diverse ps . This indicates that MixCE leads to a less noisy model distribution.

Human evaluation. Besides automatic metrics, we also conduct a human evaluation. Following Krishna et al. (2022), we conduct blind A/B testing. We randomly sample 105 examples from each dataset. For each example, we ask humans to read two generations from MLE and MixCE-finetuned GPT-2 large models, respectively, and the order of showing these two generations is random. We use unbiased sampling to get the generations. Then, we ask them to judge which one is better (or they are the same) and justify their preference, based on fluency, coherence, informativeness, and whether it is sensical. We conduct this evaluation on Amazon Mechanical Turk and collect 3 responses for each example. Please refer to Section 3.5.2 for more details and ex-

Dataset	Which is better?		
	MixCE	MLE	Same
WikiText	135*	85	95
WebText	139*	79	97
WritingPrompts	111	119	85

Table 3.5: Human evaluation results. The star (*) means significantly better ($p < 0.01$). The significance test is conducted following the bootstrap test setup (Efron and Tibshirani, 1994).

Data Size	Objective	WikiText				WebText				WritingPrompts			
		ppl	div	mauve	coh	ppl	div	mauve	coh	ppl	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
10K	MLE	29.23	0.91	0.60	0.537	22.03	0.88	0.82	0.542	30.40	0.88	0.74	0.385
	MixCE	36.70	0.88	0.93	0.546	22.79	0.83	0.86	0.562	30.65	0.87	0.81	0.395
25K	MLE	27.90	0.91	0.68	0.545	21.75	0.88	0.86	0.547	29.37	0.88	0.79	0.394
	MixCE	35.73	0.88	0.94	0.562	21.97	0.85	0.88	0.561	29.67	0.86	0.86	0.401
100K	MLE	25.93	0.90	0.69	0.559	21.31	0.87	0.90	0.556	27.63	0.87	0.88	0.401
	MixCE	34.13	0.87	0.93	0.575	21.58	0.85	0.92	0.566	28.01	0.85	0.90	0.409

Table 3.6: Unbiased sampling results of GPT-2 small models finetuned by MLE or MixCE on three datasets of different training data sizes. All metrics are the closer to the human scores the better. **Bold** numbers are the ones that are closer to human scores in each setting.

amples. The final results are shown in Table 3.5. As you can observe, MixCE-finetuned models significantly outperform MLE-finetuned models on both WikiText and WebText domains, while the two methods perform similarly on WritingPrompts. It is also worth noting that, compared to the results shown in Table 3.3, none of the 4 automatic metrics share the same trend with human evaluation.

3.4.3 Robustness & Analysis

Varying training data sizes. We test 3 other training data sizes: 10K, 25K, and 100K using GPT-2 small. Table 3.6 contains the results which share the same trend as Table 3.3: MixCE-finetuned models get worse perplexity but in general work better than MLE-finetuned models for diversity, mauve, and coherence.

Model Size	Objective	WikiText	WebText			WritingPrompts		
		c-mauve ₁₀₀	c-mauve ₁₀₀	c-mauve ₂₀₀	c-mauve ₃₀₀	c-mauve ₁₀₀	c-mauve ₂₀₀	c-mauve ₃₀₀
	Human	0.97	0.96	0.96	0.96	0.96	0.96	0.96
Small	MLE	0.92	0.93	0.92	0.90	0.94	0.94	0.92
	MixCE	0.92	0.94	0.94	0.93	0.95	0.94	0.94
medium	MLE	0.94	0.93	0.91	0.90	0.94	0.94	0.93
	MixCE	0.93	0.95	0.94	0.94	0.95	0.94	0.94
Large	MLE	0.93	0.93	0.93	0.91	0.94	0.94	0.93
	MixCE	0.93	0.94	0.94	0.93	0.95	0.95	0.95

Table 3.7: Controlled mauve results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 3.3. Human scores are not 1 because sampling 10K fragments twice result in two different sets. Each number is a 3-run average.

Model Size		WikiText	WebText			WritingPrompts		
		c-coh ₁₀₀	c-coh ₁₀₀	c-coh ₂₀₀	c-coh ₃₀₀	c-coh ₁₀₀	c-coh ₂₀₀	c-coh ₃₀₀
	Human	0.570	0.521	0.583	0.600	0.412	0.470	0.481
Small	MLE	0.504	0.444	0.515	0.535	0.350	0.412	0.429
	MixCE	0.508	0.458	0.524	0.545	0.363	0.422	0.437
Medium	MLE	0.518	0.446	0.515	0.535	0.355	0.415	0.432
	MixCE	0.527	0.484	0.546	0.565	0.362	0.425	0.437
Large	MLE	0.521	0.449	0.515	0.536	0.372	0.431	0.447
	MixCE	0.522	0.469	0.531	0.569	0.369	0.434	0.450

Table 3.8: Controlled coherence results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 3.3. Each number is a 3-run average.

Varying η and max generation length. To examine how the mixing ratio η and the max generation length affect the performance, we show the mauve score curves on the dev set in Figure 3.4. The x-axis is the mixing ratio η from 0 to 1 (MixCE=MLE when $\eta = 1$), and the y-axis is the mauve score with different max generation lengths (128, 320, and 512). First, reasonable performances are usually observed when $\eta \geq 0.1$, and only training the models with approximated reverse CE (i.e., $\eta = 0$) leads to degeneration. Second, the advantage of MixCE is more prominent when the max generation length is longer.

Controlled Mauve and Coherence. The max generation length is not the actual text length because when sampling from the model, EOS can be generated at any step. We find that the actual length of the text is a confounding factor of mauve computation. For example, when we compute mauve between a set of texts and the same set with an extra new line token after each text (or the

same set with the last k tokens being truncated), the score will be lower than 0.01. Though you may think truncating all texts to the same length can resolve this problem, we find that the *incompleteness* caused by truncation can also be a confounding factor. For instance, keeping human texts intact, we truncate texts generated by two systems by their shorter lengths (i.e., for each example, we truncate `text1` and `text2` by `min_length(text1, text2)`). Then, the system whose texts get truncated less will get a greatly larger mauve score than the other system. Therefore, to eliminate the influence of these two confounding factors, we propose a *controlled mauve* computation approach. Concretely, for the set of human texts \mathbf{T}_h and the set of model-generated texts \mathbf{T}_m , we randomly sample 10K L -length text fragments from each of these two sets. L is the number of tokens in each text fragment. After that, we compute the mauve between these two sets of 10K text fragments. We denote this controlled mauve as c-mauve_L .

$$\mathbf{F}_{h,L} = \{f_{h,L}^i\}_{i=1}^{10K}, f_{h,L}^i \sim \mathbf{T}_h$$

$$\mathbf{F}_{m,L} = \{f_{m,L}^i\}_{i=1}^{10K}, f_{m,L}^i \sim \mathbf{T}_m$$

$$\text{c-mauve}_L = \text{mauve}(\mathbf{F}_{h,L}, \mathbf{F}_{m,L})$$

To sample each fragment, we first randomly sample a text t^i from the set, and then randomly select a start token s (as long as there are more than L tokens from s to the end of t^i), then the fragment is $t^i[s : s + L]$. We set $L = 100, 200$, and 300 , except that we could not get 10K 200-token fragments from WikiText because its texts are shorter. Finally, Table 3.7 shows the results. As you can see, c-mauve scores are in general very high (≥ 0.90), which may indicate that, after controlling the confounding factors, the ability of mauve to distinguish model text from human text has been weakened. MixCE still gets better performance than MLE in most cases. The Coherence score (Su et al., 2022) computes the cosine similarity between the prompt and the continuation. We suspect that the length of the continuation may affect the score. Therefore, following the same idea of controlled mauve, we also sample 10K fragments of the same length from the set of texts for evaluation and compute coherence on the fragments. And for each fragment, we take

Model Size	Objective	WikiText	WebText	WritingPrompts
		avg. len	avg. len	avg. len
	Human	124.5	304.5	332.5
Large	MLE	114.8	284.2	325.8
	MixCE	89.0	298.9	326.4

Table 3.9: Unbiased sampling text lengths of models finetuned by MLE or MixCE on three datasets. Length is computed by simply splitting text by whitespaces.

the first 50 tokens as the prompt and the rest as the continuation. Table 3.8 shows the results. As you can observe, under this controlled setting, MixCE-finetuned models generally achieve better coherence over MLE-finetuned models.

Text length of model generations. Though by default we set the max generation length as 512, the actual text length can vary as the EOS token can be sampled at any time step. Therefore, we list the average text length of the human text and GPT2-large generations in Table 3.9. We observe that model generations are always shorter than human text. Compared to MLE, our MixCE-finetuned model produces shorter text on WikiText while producing longer text on the other two datasets. We suspect that the shorter length of MixCE on WikiText is due to the small mixing ratio (0.1) chosen based on mauve (see Table 3.11). However, we do not think shorter text length leaves to better mauve, as shown by the other two datasets and discussed in our proposal of controlled mauve.

3.5 Implementation Details

3.5.1 Best η

Table 3.10 has the best η s for synthetic experiments. Table 3.11 contains the best η s selected for GPT-2 experiments.

Model section is based on avg. js					
Vocab	Objective	Random (50%)	WebText	Random (10%)	Random (90%)
		best η	best η	best η	best η
20	Mix KLS	0.99	0.9	0.99	0.99
	JS	0.9	0.9	0.9	0.9
	MixCE*	0.99	0.99	0.99	0.99
	MixCE	0.9	0.99	0.99	0.99
50	Mix KLS	0.99	0.99	0.9	0.99
	JS	0.01	0.99	0.9	0.9
	MixCE*	0.99	0.99	0.99	0.99
	MixCE	0.99	0.99	0.99	0.9
100	Mix KLS	0.9	0.99	0.9	0.99
	JS	0.01	0.99	0.99	0.01
	MixCE*	0.99	0.99	0.99	0.99
	MixCE	0.5	0.9	0.5	0.99
500	Mix KLS	0.9	0.99	0.99	0.99
	JS	0.99	0.99	0.99	0.99
	MixCE*	0.99	0.99	0.99	0.99
	MixCE	0.1	0.5	0.1	0.1
1000	Mix KLS	0.99	0.99	0.99	0.99
	JS	0.99	0.99	0.99	0.99
	MixCE*	0.99	0.99	0.99	0.99
	MixCE	0.1	0.5	0.1	0.1

Table 3.10: The selected best η of synthetic experiments reported in Table 3.1. The model section is based on avg. js.

3.5.2 Human Evaluation Details

We conduct A/B testing (or pairwise comparison) to compare generations from two models. As shown in Figure 3.5, in each job, we give the evaluator two text paragraphs (in random order) that share the same beginning part (the prompt) but have different continuations. Then, they need to choose which one they think is better (or non-distinguishable). To avoid random selections, they are also asked to provide a justification for their choice. We find this justification not only gives us additional explanations of their choices but also helps us easily identify bad workers, because bad workers tend to use one single justification or several repeated justifications.

We instruct them by defining a good text paragraph as being:

Model section is based on mauve (max length=512) on dev set				
		WikiText	WebText	WritingPrompts
Model Size	Objective	best η	best η	best η
Small	MixCE	0.1	0.5	0.5
Medium	MixCE	0.1	0.3	0.5
Large	MixCE	0.1	0.3	0.7

Table 3.11: The selected best η of GPT-2 experiments reported in Table 3.3. The model section is based on mauve (max length=512) on the dev set.

Fluent: Should have no obviously ungrammatical sentences, missing components, etc. that make the text difficult to read.

Coherent: Should stay on topic with the prompt and build from sentence to sentence to a coherent body of information.

Informative: Should have diverse and interesting content.

Sensical: Should generally make sense.

Since short text has little information and long text is difficult to read, we only use paragraphs with 5 to 8 sentences for evaluation. If a paragraph has more than 8 sentences, we truncate it to 8 sentences. And we remove paragraphs with less than 400 or more than 2000 characters. Besides, to eliminate the influence of length difference, we do not select examples whose length difference between two paragraphs is more than 1 sentence or more than 200 characters.

We conduct this evaluation on Amazon Mechanical Turk. We only allow workers, who are located in the US, have a Masters Qualification,⁵ have an approval rate larger than 97%, and have more than 10000 HITs approved, to do our tasks. In addition, we first ran a testing batch, then manually checked the results, and selected 44 qualified workers to continue doing the rest of our tasks.

For each of the 3 datasets, we sampled 105 examples and collected 3 responses per example. In total, we received 945 human evaluations. We pay workers \$1 per response, and it takes around 5 minutes to finish one response, i.e., the hourly rate is around \$12.

⁵<https://www.mturk.com/worker/help>

Dataset	all agree	2 agree	no agreement
WikiText	24%	59%	17%
WebText	24%	52%	24%
WritingPrompts	20%	70%	10%

Table 3.12: Inter-annotator agreement. The numbers are the portions of examples that have a 3-annotator agreement (all agree), a 2-annotator agreement (2 agree), or no agreement. E.g., 24% of examples used in human evaluation for WikiText have an agreement among 3 annotators.

Table 3.12 shows the inter-annotator agreements.

3.5.3 Reproducibility

In our GPT-2 experiments, we use English text data from 3 domains: (1) WikiText (Merity et al., 2016): text from Wikipedia, and we use wikitext-103-raw-v1 from Hugging Face.⁶ Its license is Creative Commons Attribution-ShareAlike License (CC BY-SA 4.0). (2) WebText (Radford et al., 2019): text from the Web. It was used for pretraining GPT-2. The full WebText is not available but they released a subset on Github⁷. The GitHub repository contains an MIT license, and they did not specify the license of the data. But they indicated in the readme: “We look forward to the research produced using this data!” (3) WritingPrompts (Fan et al., 2018a)⁸: text from the writing prompts forum of Reddit. Its GitHub repository also contains an MIT license without specification of the data license. However, WritingPrompts has been used by many other research works, e.g., Pillutla et al. (2021). We use their official dev and test sets as much as possible. If they have fewer than 5K examples, we sample from their official training set to make up the rest.

All of our experiments were conducted on NVIDIA Tesla V100 32G GPUs. We use a single GPU to run each experiment and change the batch size to fit models of different sizes. When fine-tuning GPT-2 small using a single GPU with MLE or MixCE, it took less than 1 hour to finish

⁶<https://huggingface.co/datasets/wikitext>

⁷<https://github.com/openai/gpt-2-output-dataset>

⁸<https://github.com/facebookresearch/fairseq/tree/main/examples/stories>

5 epochs on 50K WikiText training data and took less than 2 hours to finish 5 epochs on 50K WebText or WringPrompts training data.

We implemented our GPT-2 based models based on the GPT-2 modeling code from Hugging Face Transformers⁹. For training and evaluation, we modified the example script of causal language model training¹⁰. We used the default optimizer, learning rate, scheduler, etc. in that script. But we set the maximum training epochs as 5 and changed the batch size and gradient accumulation steps based on the model size to fit it in one 32G-memory GPU.

3.6 Conclusion

We propose a novel training objective, MixCE, for autoregressive language modeling. MixCE combines forward and reverse cross-entropies, which can be viewed as combining two complementary driving forces for better fitting the model distribution to the data distribution. We demonstrate the superiority of MixCE over MLE in both synthetic and real settings via both automatic and human evaluations. In the future, MixCE can be potentially used for pretraining language models.

⁹https://github.com/huggingface/transformers/blob/main/src/transformers/models/gpt2/modeling_gpt2.py

¹⁰https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_clm_no_trainer.py

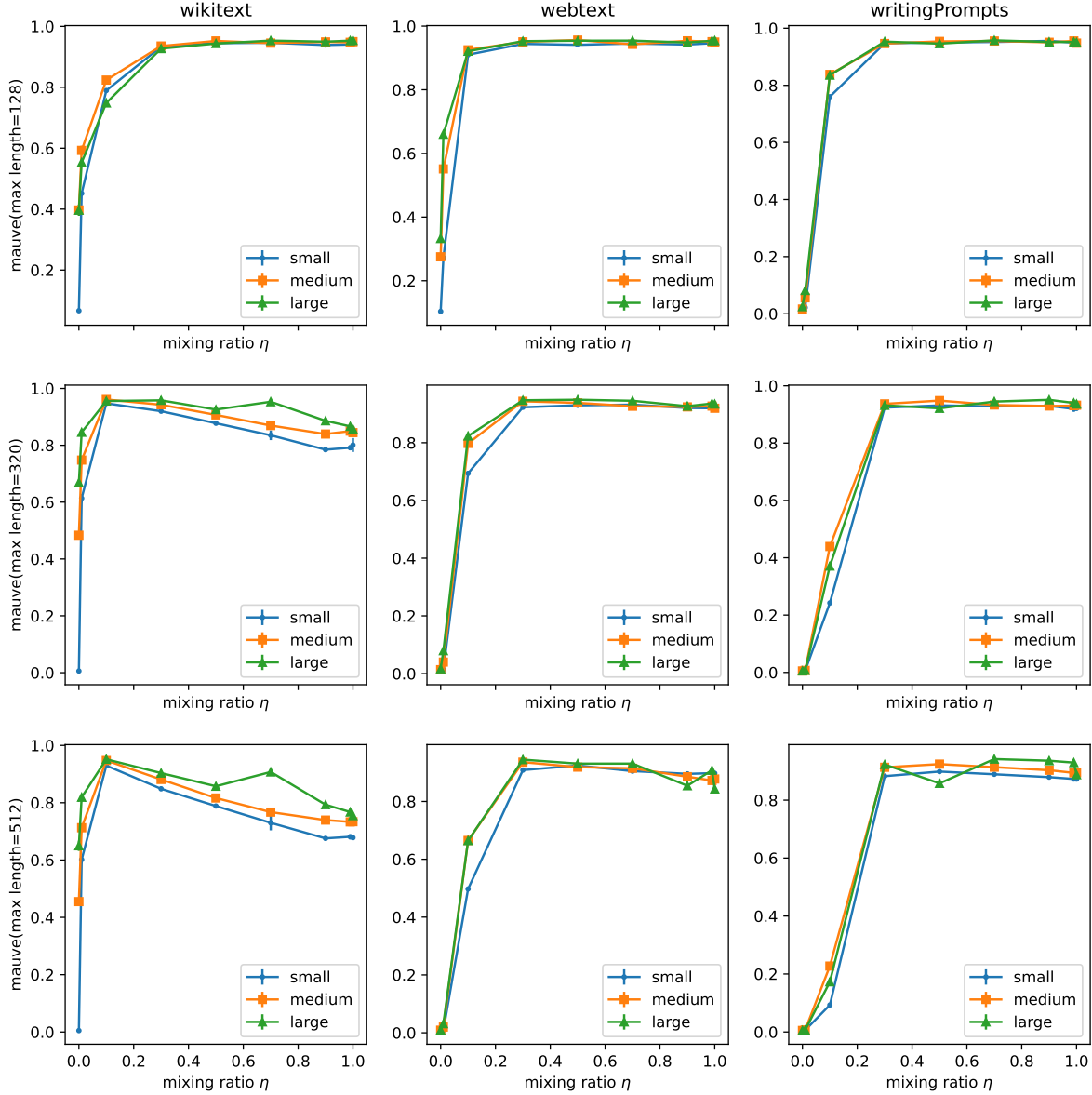


Figure 3.4: The mauve scores obtained by MixCE-finetuned GPT-2 models on development sets with different max generation lengths and different η . Note that when $\eta = 1$, MixCE is equivalent to MLE. The x-axis is the mixing ratio η , and the y-axis refers to mauve scores with different max generation lengths. The 3 lines in each subplot show the results of GPT-2 models in different sizes. The 3 subplots in each row are the results of 3 datasets respectively. Unbiased sampling is used as the decoding method. Each dot is the average of 3 runs of sampling and the error bar shows the standard deviation of 3 runs.

Instructions (Please read carefully to ensure that your work gets approved as quickly as possible!)

Welcome!

We need your help to compare the quality of two paragraphs of text.

For each assignment, you will be given **two paragraphs** of the same topic. They share the same **beginning part (or the prompt)**. Your task is to judge **which one is better and justify your choice**. A good text paragraph is:

1. **Fluent:** Should have no obviously ungrammatical sentences, missing components, etc. that make the text difficult to read.
2. **Coherent:** Should stay on topic with the prompt and build from sentence to sentence to a coherent body of information.
3. **Informative:** Should have diverse and interesting content.
4. **Sensical:** Should generally make sense.

Please note that:

1. It is NOT necessary that one paragraph is better than the other. You should choose "non-distinguishable" if you think they are both very poor/good.
2. Paragraphs may contain facts that you are not familiar with. You do NOT need to verify those facts because **factuality is not a concern in this evaluation**.
3. Paragraphs are excerpts from longer text, so the ending part may not be complete. But **incompleteness should NOT affect your judgment**.
4. When justifying your choice, please be **as concrete as possible and point to problems** in the paragraphs.

Paragraph-1:

[WP] A new drug let's you live a lifetime in one dream 30 minutes for an entire life? Wonderful, just pop the pill before you go to bed and when you reach peak REM, you'll enter a whole new life, one you never could have taken apart. But insert those 3 or 4 pills and the life begins to unravel again. So i took 3 pills that i got from carnival a few weeks ago. It was cheap and stuff boring but i thought it might help me sleep freshman year of high school. Ill hangrily pill suddenly and go back to my dream. Unfortunately the next thing i know my grandfather has died. I start walking up my stairs and checking every room for a note, and some kind of family photograph. One by one I reach my room, and find nothing but faded pictures of my grandparents, then a wrong number on the phone app

Paragraph-2:

[WP] A new drug let's you live a lifetime in one dream 30 minutes for an entire life? Wonderful, just pop the pill before you go to bed and when you reach peak REM, you'll enter a whole new life, one you never knew existed until just now. The bright light blipped and my eyes dropped open as the room automatically blacked out. A weak light throbbbed in my right temple, and another flowed back into the darkness. Winds rolled through the lower levels of the dormitory's cluster of rooms, and fresh snow drifted across the windowless, cracked rafters. I heard a sudden flare of static, and then silence. I strode onto the roof and jumped off like a springsteen with a single, softly fading tune. My hands began to tremble, and my feet began to throb. I scrambled to the breakfast nook, sat on the roof bench, and grabbed the nearest lamp.

Which paragraph do you think is better?

Paragraph-1

Paragraph-2

Non-distinguishable

Please justify your preference:

I think they're equally 'informative,' but Paragraph 2 wins on fluency, coherence, and comes far closer to making sense. Paragraph 1 is basically a heaping helping of word salad.

Figure 3.5: Human evaluation interface and a random example from our collected human annotations.

CHAPTER 4: SEMI-AUTOMATIC SUMMARY EVALUATION

4.1 Introduction

Evaluating the quality of summaries is a challenging task. Human evaluation is usually regarded as the gold standard. Out of different human evaluation methods, *Pyramid* (Nenkova and Passonneau, 2004) has been perceived as an objective and reliable protocol and used by early summarization benchmarks, e.g., TAC (DBL, 2008, 2009). Given one or several reference summaries of an example, human assessors first exhaustively extract Summary Content Units (SCUs), each SCU contains a single fact, from the reference(s), and then check whether they are present in a system summary. Figure 4.1 shows an example of human-labeled SCUs. Despite the reliability, manual evaluation is usually: (1) *not reproducible*, results may change when different evaluators are involved, making it hard to compare the results across papers; (2) *expensive*, in terms of time and cost. Thus, it is unlikely to apply human evaluation extensively in model selection (e.g., to choose the best checkpoint); instead, people usually treat it as an additional quality verification step. Aiming to work as a proxy of humans, many automatic metrics have been proposed (Lin, 2004; Tratz and Hovy, 2008; Giannakopoulos and Karkaletsis, 2011; Yang et al., 2016; Zhang et al., 2020c; Deutsch et al., 2021). However, most of them cannot reliably substitute human evaluation due to the unstable performance across datasets (Bhandari et al., 2020), weak to moderate correlations with human judgment (Fabbri et al., 2021), or more indication of topic similarity than information overlap (Deutsch and Roth, 2021).

In this work, we want to combine human and automatic evaluations and find a balance between reliability and reproducibility (plus expense). Recall the Pyramid method (Nenkova and Passonneau, 2004), where these SCUs for reference summaries only need to be annotated once, then they can be fixed. It means SCUs can come with the datasets and are reusable for evaluating

different systems. Hence, what hinders this method from being reproducible is its second step of asking humans to judge the presence of SCUs in system summaries. Whenever we have a new summarizer, we need to collect human labels for this step. Therefore, we propose to retain the reusable SCUs but replace human effort in the second step with a neural model. Basically, people are answering *whether a SCU is entailed by the summary*, which is closely related to the Natural Language Inference (NLI) task, i.e., judging whether a hypothesis is entailed by the premise. A lot of NLI datasets are available (Bowman et al., 2015; Williams et al., 2018a; Thorne et al., 2018; Nie et al., 2020) and recent NLI models have achieved close-to-human-level performance. Hence, we use a pretrained NLI model and finetune it on some in-domain gold labels of SCUs’ presence. Then, we replace humans with the finetuned model, so that the evaluation results are reproducible as long as the same model is used. Meanwhile, it can run automatically during development to guide model selection and the evaluation cost will be dramatically reduced. Shapira et al. (2019) propose *LitePyramid* to simplify the standard Pyramid method via crowdsourcing. Following but different from their work, we additionally automate the presence annotation, and hence we call our method *Lite²Pyramid*.

Lite²Pyramid still requires human efforts to extract SCUs from reference summaries, and this step is usually considered to be more difficult. Early benchmarks, e.g., TAC (DBL, 2008, 2009), are small-sized with fewer than 100 examples in the evaluation set, for which it is already expensive to manually collect SCUs. However, current popular summarization datasets, e.g., CNN/DM (Hermann et al., 2015), contain more than 10K evaluation examples, and hence we want to simulate SCUs via an automatic method for such large-scale datasets. For this, we make use of Semantic Role Labeling (SRL) that can automatically decompose a sentence to semantic triplets, e.g., *subject-verb-object*, and we take each triplet as a pseudo-SCU, which we call Semantic Triplet Unit (STU). Figure 4.1 illustrates the difference between SCUs and STUs. Although STUs do not always contain a single fact and some information might also be misrepresented, we find that it can reasonably simulate SCUs and lead to a fully automatic metric, *Lite³Pyramid*.

Lastly, instead of using either all human-labeled SCUs or all automated STUs, we investigate balanced trade-offs in between, e.g., using half SCUs and half STUs. A naive way is to randomly sample some reference sentences and substitute their SCUs with STUs. However, we find it is unstable and sometimes even works worse than using all STUs. More reasonably, we design an *active learning* (Settles, 2012) inspired selection method to help decide which sub-parts of the dataset are more worthy of obtaining expensive SCUs for. For this, we develop a regressor to predict the “simulation easiness” of each reference sentence: if a sentence is too complex to be well represented by STUs, we will ask humans to annotate SCUs for it; otherwise, we can apply automatic SRL. We call this method as *Lite^{2.x}Pyramid*, since it provides a smooth, flexible transition from *Lite²Pyramid* to *Lite³Pyramid* and balances reliability with cost.

To comprehensively evaluate the quality of metrics, we not only use 3 existing meta-evaluation datasets (TAC2008 (DBL, 2008), TAC2009 (DBL, 2009), REALSumm (Bhandari et al., 2020)) but also newly collect *PyrXSum* with 100 XSum (Narayan et al., 2018a) test examples plus summaries produced by 10 systems. Next, we compare our new metrics to 15 existing automatic metrics on these 4 meta-evaluation setups for both *system-level* and *summary-level* correlations with human Pyramid scores. We find that *Lite²Pyramid* consistently has the best summary-level correlations and is reliable as an out-of-the-box metric. *Lite³Pyramid* also mostly performs better or competitively. Lastly, the regressor-based *Lite^{2.x}Pyramid* can help substantially reduce annotation efforts for only small correlation drops, e.g., on TAC2008, TAC2009, it trades off only 0.01 absolute summary-level Pearson correlation and 0 system-level correlation for 50% SCU reduction.

Github repository: <https://github.com/ZhangShiyue/Lite2-3Pyramid>

4.2 Background and Related Work

Each example in a summarization dataset contains single or multiple source document(s) and one or several human-written reference(s). System-generated summaries are evaluated by com-

paring them to the references (i.e., reference-based) or directly scored (i.e., reference-free). This evaluation process is critical and directly affects our development choices.

Human (or manual) evaluation has been considered as the gold standard. Early benchmarks (DBL, 2008, 2009) conducted three human evaluations: *Responsiveness*, *Linguistic Quality*, and *Pyramid*. The first two ask humans to directly rate the overall responsiveness or linguistic quality on a Likert scale. Following this, some works collect ratings for different aspects, e.g., relevance, readability (Paulus et al., 2018; Kryscinski et al., 2019; Fabbri et al., 2021). However, these ratings may suffer from raters’ subjectivity. Pyramid (Nenkova and Passonneau, 2004) has been perceived as a more objective method, and it is reference-based. It has two steps: *pyramid creation* and *system evaluation*. In the first step, humans exhaustively find the Summary Content Unit (SCU) contributors from references, each contributor describes a single fact; contributors with the same meaning will be merged into one single SCU; then each SCU is weighted by how many contributors it has, equal to the number of references in which it is found. In the second step, each SCU has been manually checked its presence in the system summary; and the Pyramid score is the normalized sum of present SCUs’ weights (essentially, a recall score). Passonneau (2010) normalize it by the total weight of the best possible summary. Recently, Shapira et al. (2019) propose *LitePyramid*. It removes SCU merging and weighting, allowing SCUs of the same meaning to co-exist, and they show that the evaluation can be reliably conducted by crowdsourcing workers.

Automatic metrics trade off the reliability of human evaluation for reproducibility, low cost, and fast speed. Many automatic metrics have been introduced, the majority of which are reference-based. Some metrics measure the n-gram overlap (Papineni et al., 2002; Lin, 2004), out of which ROUGE (Lin, 2004) is the most widely adopted metric till today. Some other works compute the similarity over n-gram graphs (Giannakopoulos and Karkaletsis, 2011; Giannakopoulos et al., 2008) or distributions (Lin et al., 2006). Since exact n-gram matching is too rigid, ME-TEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014) provides flexibility by stemming, synonyms, etc., and recently, a few metrics enable “soft” matching through contextualized

word embeddings (Zhao et al., 2019; Clark et al., 2019; Zhang et al., 2020c). However, Deutsch and Roth (2021) point out that the n-gram based metrics indicate more topic similarity than information overlap. Structural evaluation metrics have also been proposed beyond n-grams. BEwT-E (Tratz and Hovy, 2008) decomposes the system summary and the reference(s) into syntactic units and compute their similarities, and decomposed-ROUGE (Deutsch and Roth, 2021) computes ROUGE for each syntactic category. APES (Eyal et al., 2019) and QAEval (Deutsch et al., 2021) are QA-based metrics that assume similar answers will be obtained from similar system summaries and reference(s).

Automatic Pyramid methods have also been proposed (Yang et al., 2016; Hirao et al., 2018; Gao et al., 2019). They usually decompose both the system summary and the references into smaller units (e.g., Elementary Discourse Units) and compare the two list of units. Differently, our *Lite³Pyramid* only decomposes the reference summaries to semantic triplet units (STUs), and we use NLI to judge the presence of each STU in the system summary, which is closer to the original Pyramid’s procedure and leads to better correlations with human scores (refer to Section 4.5). Peyrard et al. (2017) propose a learned metric, S3, that is trained to directly predict human Pyramid or Responsiveness scores based on ROUGE, FrameNet features, etc. Sellam et al. (2020) propose a learned metric for machine translation, BLEURT, that finetunes a BERT (Devlin et al., 2019) model with human ratings to directly predict the similarity score of a (reference, model translation) pair, and they show that it can also be successfully applied for WebNLG (Gardent et al., 2017) tasks. We are similar to both S3 and BLEURT in the way of learning to evaluate through finetuning NLP models with human labels. Xu et al. (2020c) is distantly related to us in the way of representing texts by SRL, but it is used to weigh the content in the source document(s). Besides, some reference-free metrics are introduced for summary quality estimation (Xenouleas et al., 2019; Gao et al., 2020; Vasilyev et al., 2020) or faithfulness evaluation (Durmus et al., 2020; Wang et al., 2020a).

Semi-automatic evaluation is introduced by Zhou et al. (2007). They automatically decompose both system summary and reference(s) into semantic units and then ask humans to

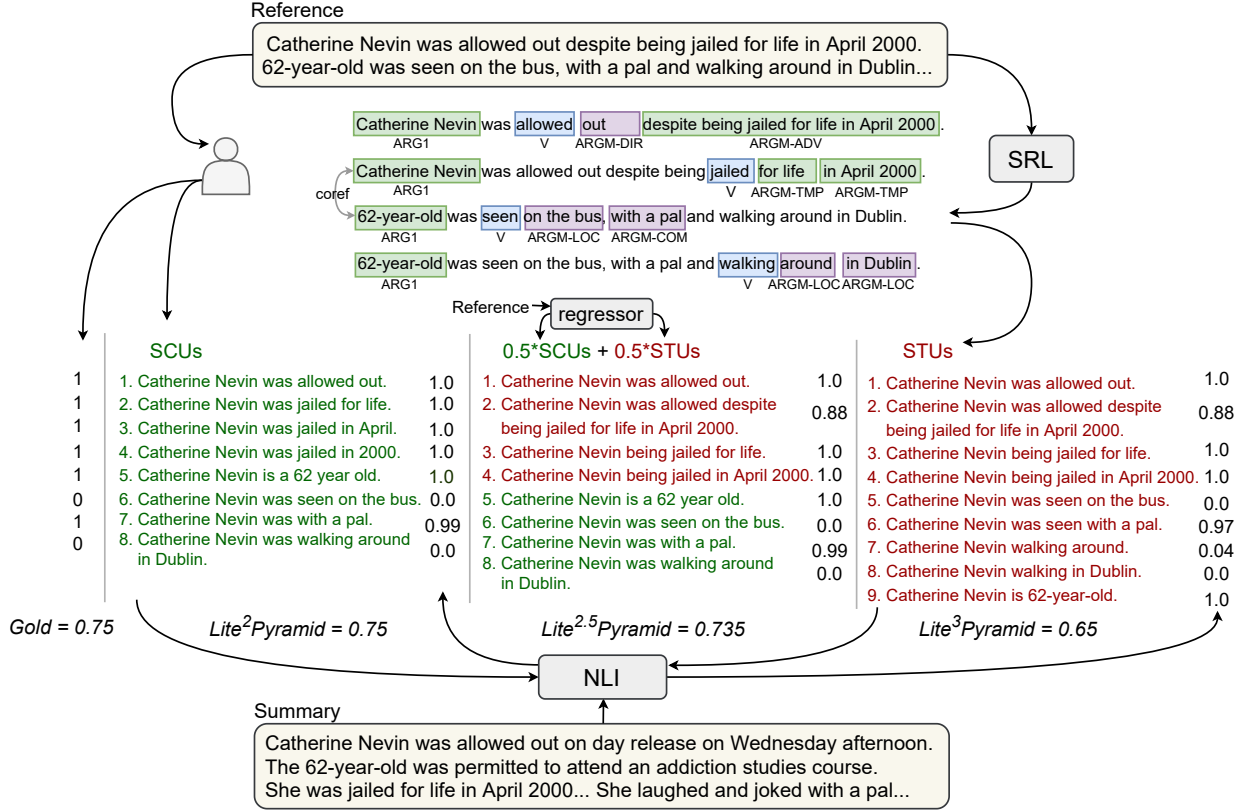


Figure 4.1: The illustration of our metrics. This data example is from REALSumm (Bhandari et al., 2020) (we omit unnecessary content by ‘...’). For gold labels, ‘1’ stands ‘present’ and ‘0’ stands ‘not present’. Other scores are the 2-class entailment probabilities, $p^{2c}(e)$, from our fine-tuned NLI model.

match/align the two lists of units. In contrast, our semi-automatic Lite²Pyramid retains the reusable SCUs while automatically judges the SCUs’ presence in the system summary (via NLI).

4.3 Our Method

4.3.1 Lite²Pyramid

Lite²Pyramid is a semi-automatic metric that retains human-labeled Summary Content Units (SCUs) to represent reference summaries of a data example i , i.e., $\{SCU_{ij}\}_{j=1}^{N_i}$, where N_i is the total number of SCUs from all reference summaries. The original Pyramid (Nenkova and Passonneau, 2004; Passonneau, 2010) assumes there are multiple references available (e.g., TAC datasets (DBL, 2008, 2009) have 4 references per example). Therefore, each SCU comes with

weight, $\{w_{ij}\}_{j=1}^{N_i}$, representing the number of reference summaries in which the SCU is found.

To evaluate a particular system summary s_i , the standard Pyramid method manually checks each SCU’s presence, sums up the weights of present SCUs, and normalizes it:

$$\text{Pyramid}_i = \frac{\sum_{j=1}^{N_i} w_{ij} \text{Presence}(SCU_{ij}, s_i)}{\text{the best possible score}} \quad (4.1)$$

The best possible score is the highest sum of weights the summary can obtain with the same number of present SCUs (details can be found in (Passonneau, 2010)). Differently, LitePyramid (Shapira et al., 2019) takes a union of SCUs from all reference summaries *with duplication* (we use SCU^* to distinguish it from the de-duplicated SCU used above) and then samples the same number (K) of SCUs for every data example, hence:

$$\text{LitePyramid}_i = \frac{\sum_{j=1}^K \text{Presence}(SCU_{ij}^*, s_i)}{K}$$

Without weighting, this method also works in single-reference situations. Different from this method, we keep the exhaustive set (instead of a fixed-size sample) of SCUs for each example (also used by Bhandari et al. (2020)). Importantly, we replace human efforts of checking SCUs’ presence with a Natural Language Inference (NLI) model f_{nli} ’s entailment prediction. Using e to denote entailment, our metric can be written as:

$$\text{Lite}^2\text{Pyramid}_i = \frac{\sum_{j=1}^{N_i} w_{ij} f_{\text{nli}}(e|SCU_{ij}, s_i)}{\sum_{j=1}^{N_i} w_{ij}} \quad (4.2)$$

Note that multiplying the weights and dividing by the sum of the weights is equal to repeating SCU_i for w_i times, which shows how we treat SCUs as an exhaustive set with duplication. For single-reference datasets (CNN/DM or XSum), the weights are all 1. Plus, the above equations all compute summary-level scores. To get one single score for the system, we simply take the average across examples, e.g., $\frac{1}{|D|} \sum_{i \in D} \text{Lite}^2\text{Pyramid}(s_i)$.

The f_{nli} function can be implemented in four different ways, denoted as p^{3c} , l^{3c} , p^{2c} , l^{2c} , and explained below. Following the standard 3-class setting of NLI tasks, the NLI model will predict whether the SCU_{ij} is entailed by or neutral to or contradicted with the summary s_i . Hence, we can use either the output probability of entailment class $p^{3c}(e)$ or the predicted 1 or 0 entailment label $l^{3c}(e)$ as the function f_{nli} . However, existing NLI datasets (Bowman et al., 2015; Williams et al., 2018b; Thorne et al., 2018; Nie et al., 2020) have different data distributions and domains from the summarization data; hence models trained on these datasets may not perform well in judging the presence of SCUs. Therefore, we finetune the pretrained NLI model by human-labeled SCUs plus presence labels. Since humans only give 2-class labels (present or not present), we adapt the model to perform two-way classification. Specifically, we add up the logits of neutral (n) and contradiction (c) classes as the logit of the “not present” label: $p^{2c}(e) = \frac{\exp(\text{logit}_e)}{\exp(\text{logit}_e) + \exp(\text{logit}_n + \text{logit}_c)}$. Again, we can use $p^{2c}(e)$ or $l^{2c}(e)$ as f_{nli} after finetuning. In our experiments, we call the pretrained NLI model on NLI datasets as “zero-shot” because it has not seen summarization data. Empirically, we find that when using the zero-shot NLI model, l^{3c} works best; while after finetuning, p^{2c} usually works best.

4.3.2 Lite³Pyramid

Lite³Pyramid fully automates Lite²Pyramid by simulating the human-annotated SCUs with automatic extracted semantic triplets. We use a Semantic Role Labeling (SRL) model (Carreras and Màrquez, 2005; Palmer et al., 2010; He et al., 2017; Shi and Lin, 2019) to achieve this goal. SRL determines the latent predicate-argument structure of a sentence, e.g., *who did what to whom*. As shown in Figure 4.1, the SRL model will identify several frames for each sentence, and each frame has one verb and a few arguments. For each frame, we keep the verb and any arguments before the verb unchanged, then we enumerate the arguments after the verb to form a list of triplets as $\{(\text{ARG}_{\text{before}}, \text{V}, \text{ARG}_{\text{after}}^i)\}_{i=1}^M$, where M is the number of arguments after the verb. We concatenate the three elements in each triplet to form a short sentence because a SCU is a short sentence and we want to resemble it as much as possible. We call these short sentences Se-

semantic Triplet Units (STUs).¹ For example, as illustrated by Figure 4.1, based on the 4 frames identified by SRL, we extract 9 STUs from the reference.

Since one entity can be referred to by pronouns or different names in the summary, we also apply Coreference Resolution (Lee et al., 2018) to improve the simulation quality. As shown in Figure 4.1, *Catherine Nevin* and *62-year-old* are identified as coreference, so we use *Catherine Nevin* as the subjects of STUs and add an additional STU *Catherine Nevin is 62-year-old*.² In our experiments, we only apply coreference resolution for REALSumm because empirically, on TAC datasets, we find applying it works worse than not applying; and PyrXSum has one-sentence summaries where coreference hardly appears.³ Although STUs seem to reasonably simulate SCUs for the example in Figure 4.1, it has limitations, especially, when the sentence is syntactically complicated, e.g., with a lot of modifiers, clauses, complements (refer to Section 4.5 for more discussions).

After we obtain the STUs from all reference summaries, we score a system summary s_i by:

$$\text{Lite}^3\text{Pyramid}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} f_{\text{nli}}(e | \text{STU}_{ij}, s_i)$$

where M_i is the total number of STUs. Note that there is no weight because we extract STUs from every reference summary and take a union, which allows STUs of the same meaning to co-exist.

¹Note that simple concatenation might not lead to grammatical sentences, but we expect the NLI model to be robust to small grammar errors. Additionally, we make a small fix in two cases: if the token before V in the original sentence is classified as a negation modifier, ARG-M-NEG, or is a *Be* verb, we add it to the STU sentence (e.g., for the 3rd STU in Figure 4.1, we bring back “being” before “jailed”).

²In practice, we use the name appeared first in the reference to unify the mentions in STUs and use the template “name₁ is name_n” to generate additional STUs.

³Even for REALSumm, removing the coreference resolution step will only cause around 0.01 absolute correlation drops.

4.3.3 Lite^{2.x}pyramid

As discussed so far, human-annotated SCUs are accurate yet expensive, whereas automatically extracted STUs are cheap yet sometimes erroneous. The next natural question is how to find a balance between them. One way is to randomly replace 50% sentences' SCUs with STUs, but a more intuitive way is to make the decision based on the "easiness" of simulating the sentence's SCUs by STUs. If the sentence is unlikely to be well represented by STUs, we can ask humans to label SCUs for it; otherwise, we can use STUs to reduce cost. This is similar to how *active learning* (Settles, 2012) chooses which training examples to collect human labels for. We define simulation easiness as the average simulation accuracy of each SCU. ROUGE-1-F1 ($R1_{F1}$) (Lin, 2004) is used to measure the simulation accuracy: $Acc_j = \max_m R1_{F1}(SCU_j, STU_m)$. Then, the easiness of a sentence with N_{sent} SCUs is written by $Easiness_{sent} = \frac{1}{N_{sent}} \sum_{j=1}^{N_{sent}} Acc_j$. The higher the easiness score is, the more accurately the STUs resemble SCUs.

After we obtain these gold easiness scores, we want to train a regressor to predict the score based on sentence complexity features. As we mentioned above, the sentence's syntax can indicate its simulation difficulty. Therefore, we get the Constituency Parsing tree (Joshi et al., 2018) of each sentence and define the following features: (1) sentence length; (2) linearized parsing tree length; (3) parsing tree depth; (4) sentence length / parsing tree depth; (5) the counts for each of the 65 non-terminal tokens (e.g., NNP). In total, we represent each sentence with a 69-dim feature vector. Then, we train an XGBoost (Chen and Guestrin, 2016) regressor to predict the simulation easiness by minimizing the mean squared errors. Given this regressor, we propose to replace top $0.x$ scored sentences' SCUs with STUs, leading to Lite^{2.x}Pyramid. For example, Lite^{2.5}Pyramid (illustrated in Figure 4.1) means that we use STUs for the top 50% scored sentences and use SCUs for the other half.

4.4 Evaluation

4.4.1 Correlation with Human Scores

Following the standard meta-evaluation strategies used in previous works (Peyrard et al., 2017; Bhandari et al., 2020; Deutsch et al., 2021), we evaluate metrics by two types of correlation with gold human scores.

System-level correlation aims to evaluate *how well the metric can compare different summarization systems?* We denote the correlation measure as K , human scores as h , the metric as m , and generated summaries as s . We assume there are N examples and S systems in the meta-evaluation dataset. Then, the system-level correlation is defined as:

$$K_{m,h}^{sys} = K\left(\left[\frac{1}{N} \sum_{i=1}^N m(s_{i1}), \dots, \frac{1}{N} \sum_{i=1}^N m(s_{iS})\right], \left[\frac{1}{N} \sum_{i=1}^N h(s_{i1}), \dots, \frac{1}{N} \sum_{i=1}^N h(s_{iS})\right]\right)$$

Summary-level correlation answers *if the metric can reliably compare summaries generated by different systems for the same document(s).* Using the same notations, this correlation is written by:

$$K_{m,h}^{sum} = \frac{1}{N} \sum_{i=1}^N K\left([m(s_{i1}), \dots, m(s_{iS})], [h(s_{i1}), \dots, h(s_{iS})]\right)$$

We use Pearson r or Spearman ρ as the correlation measure K . Pearson measures linear correlation while Spearman measures ranking correlation.

4.4.2 Metrics for Comparison

ROUGE-1, **ROUGE-2**, and **ROUGE-L** (Lin, 2004) are based on n-gram overlap and are widely used in summarization literature till today.

AutoSummENG (Giannakopoulos et al., 2008) uses n-gram graphs to compare the system summary to the reference(s).

METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014) computes similarity through text alignment and uses stem, synonyms, paraphrases to allow more flexible matching.

BEwT-E (Tratz and Hovy, 2008) decomposes summary into syntactic units and computes the similarity based on those units.

S3 (Peyrard et al., 2017) is a learned metric trained on TAC2008/2009 datasets to predict human Pyramid (**pyr**) or Responsiveness (**resp**) scores.

PyrEval (Gao et al., 2019) automate Pyramid by simulating SCUs through Emergent Discovery of Units of Attraction. It returns four scores. Empirically, we find that **quality** and **comprehensive** work better, so we only keep these two in our result tables. Note that it only supports multi-reference situations because it retains SCUs’ weighting step.

BERTScore (Zhang et al., 2020c) aligns unigrams between two texts through the contextualized word embeddings from BERT (Devlin et al., 2019). We also compare to **BERTScore (idf)** that down-weights unigrams with high document frequency.

MoverScore (Zhao et al., 2019) also uses contextualized word embeddings. Differently, they minimize the “transportation cost” between two texts.

QAEval (Deutsch et al., 2021) leverages Question Answering to evaluate the similarity of two texts, i.e., if they have the same meaning, the same answer should be inferred from them for the same question. They use either Exact Match (**EM**) or F1 (**F1**) to evaluate answer similarity.

4.4.3 Data

We evaluate human-metric correlations on three existing English meta-evaluation datasets: *TAC2008* (DBL, 2008), *TAC2009* (DBL, 2009), *REALSumm* (Bhandari et al., 2020). TAC08

contains 96/58 examples/systems and TAC09 has 88/55 examples/systems. We compute the correlations with their official Pyramid scores (Equation 4.1).⁴ REALSumm has 100 CNN/DM (Hermann et al., 2015) test examples and 25 systems. They label SCUs by themselves and collect SCU-presence labels on Amazon Mechanical Turk (AMT). Both TAC and CNN/DM have long and extractive summaries. To complete our evaluation, we newly collect an English meta-evaluation dataset *PyrXSum* for 100 XSum (Narayan et al., 2018a) (has short and abstractive summaries) testing examples. Following REALSumm, we (authors) manually label SCUs and collect SCU-presence labels for summaries generated by 10 systems⁵ on AMT. We collect 4 responses per summary (100 * 10 * 4 HITs) and filter responses from a noisy worker. We use the majority vote to label each SCU’s presence and break ties by “not present” . See more data collection details of *PyrXSum* in Section 4.6.1.

4.4.4 Models

We use the pretrained RoBERTa-large (Liu et al., 2019b) based NLI model released by Nie et al. (2020), which has been trained on multiple NLI datasets. We continually finetune this model with the gold SCUs plus SCU-presence labels always for 2 epochs. For SRL, Coreference Resolution, and Constituency Tree Parser, we use the out-of-the-box tools provided by AllenNLP (Gardner et al., 2018; Shi and Lin, 2019; Lee et al., 2018; Joshi et al., 2018). See the complete implementation details in Section 4.6.2.

Metrics	System-level								Summary-level							
	TAC08		TAC09		RealSumm		PyrXSum		TAC08		TAC09		RealSumm		PyrXSum	
	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ
ROUGE-1	.87	.87	.91	.86	.82	.83	.92	.90	.62	.61	.69	.63	.53	.50	.52	.50
ROUGE-2	.90	.90	.92	.90	.84	.82	.93	.91	.63	.62	.71	.64	.46	.43	.53	.51
ROUGE-L	.87	.86	.93	.87	.83	.81	<u>.94</u>	.92	.57	.55	.66	.59	.46	.42	.52	.51
AutoSummENG	.90	.89	.91	.89	.53	.51	.92	.91	.65	.64	.71	.64	.34	.34	.56	.53
METEOR	.90	.89	.93	.88	.84	.84	<u>.94</u>	.89	.65	.64	.73	.68	.54	.49	<u>.58</u>	<u>.56</u>
BEwT-E	.92	<u>.91</u>	.95	.92	.83	.84	.93	.86	.66	.65	.75	.68	.47	.45	.54	.52
S3 pyr	.90	.89	.95	.89	.86	.85	<u>.94</u>	.89	.66	.65	.75	.68	.54	.50	.57	.54
S3 resp	.91	.91	.94	.90	.86	.86	<u>.94</u>	.90	.67	.65	.74	.68	.52	.48	.57	.54
PyrEval qual	.83	.81	.88	.80	-	-	-	-	.40	.39	.49	.44	-	-	-	-
PyrEval comp	.83	.80	.90	.79	-	-	-	-	.41	.40	.53	.45	-	-	-	-
BertScore	.88	.87	.90	.90	.73	.77	.92	.89	.61	.60	.70	.65	.48	.46	.57	.54
BertScore (idf)	.89	.88	.91	.90	.73	.78	.93	.90	.62	.61	.71	.66	.48	.46	<u>.58</u>	.55
MoverScore	.91	.89	.95	.90	.40	.31	.92	.91	.64	.63	.73	.68	.39	.36	.57	.54
QAEval EM	.83	.81	.85	.83	.61	.51	.86	.85	.48	.48	.64	.55	.28	.27	.29	.27
QAEval F1	.89	.87	.90	.87	.72	.65	.90	.83	.61	.60	.70	.63	.38	.35	.46	.42
Lite ³ Pyramid	<u>.93</u>	<u>.91</u>	.97	<u>.93</u>	<u>.89</u>	<u>.87</u>	.89	.86	<u>.71</u>	<u>.69</u>	<u>.78</u>	<u>.73</u>	<u>.57</u>	<u>.53</u>	.51	.48
Lite ^{2.5} Pyramid	.95	.93	.97	.94	.90	.88	.92	.87	.76	.75	.82	.77	.62	.57	.64	.59
Lite ² Pyramid	.95	.93	.97	.94	.89	.86	.95	.92	.77	.76	.83	.78	.64	.60	.74	.66
Lite ² Pyramid-0	.86	.83	.95	.88	.86	.82	.96	.92	.62	.61	.74	.68	.56	.53	.73	.72

Table 4.1: 5-fold (split by examples) cross-validation results. In each column, the **bold** numbers are the best and the underline numbers are the best out of automatic metrics. All Lite²Pyramid-0 numbers are based on $f_{\text{nli}} = l^{3c}$, while all other numbers of our metrics are based on $f_{\text{nli}} = p^{2c}$.

4.5 Results

4.5.1 Human-Metric Correlation Results

Since we find that finetuning the NLI model with in-domain presence labels is greatly beneficial, following Peyrard et al. (2017), we evaluate by 5-fold cross-validation. For each dataset, we split it into 5 folds, finetune the NLI model and train the regressor on 4 folds, test on the left one, and repeat for 5 times. We report the 5-fold average correlations of both our metrics and the 15 metrics we compare to for fair comparison. Instead of random splitting, we split the data *by examples* or *by systems*, aiming to check the generalizability across examples or systems. E.g.,

⁴We find that the exhaustive set based computation (replacing f_{nli} in Equation 4.2 by gold labels) has close to perfect correlation with TAC’s official scores. REALSumm also use this computation as reflected by the gold score in Figure 4.1.

⁵Fast Abs RL (Chen and Bansal, 2018), PtGen (See et al., 2017), ConvS2S and T-ConvS2S (Narayan et al., 2018a), TransAbs and BertAbs and BertExtAbs (Liu and Lapata, 2019), T5 (Raffel et al., 2020), BART (Lewis et al., 2020a), PEGASUS (Zhang et al., 2020a)

if we split REALSumm by examples, each fold has summaries of 20 examples; when split by systems, each fold has summaries generated by 5 systems.

Table 4.1 shows our 5-fold (split *by examples*) cross-validation results. Firstly, it can be observed that our **Lite²Pyramid** always has the best or close to the best correlations; especially, it has 0.08 to 0.16 higher summary-level correlations than the best metrics we compare to. It demonstrates the advantage of semi-automatic evaluation which dramatically improves reliability without losing reproducibility. Meanwhile, it indicates that the finetuned NLI model can generalize to new data examples and works reasonably well as a proxy of human judgment. In contrast, Lite²Pyramid-0, which uses a non-finetuned NLI model, usually works greatly worse than Lite²Pyramid, which indicates the importance of in-domain finetuning. It is surprising that Lite²Pyramid-0 works better than or similar to Lite²Pyramid on PyrXSum. We conjecture that because our PyrXSum is relatively small-size, the finetuning will not make big difference.

Secondly, our **Lite³Pyramid** has the best correlations comparing to the other automatic metrics, except for PyrXSum; again, its advantage is more prominent on summary-level correlation (around 0.03 to 0.05 better). Its failure in PyrXSum is caused by the limitation of SRL. XSum’s reference summary sentences usually have a lot of modifiers, adverbial phrases/clauses, or complements, which increases the difficulty of decomposing it into STUs. E.g., for the summary *“Netherlands midfielder Wesley Sneijder has joined French Ligue 1 side Nice on a free transfer”*, human annotates the following 5 SCUs: *“Wesley Sneijder is a midfielder”*, *“Wesley Sneijder comes from Netherlands”*, *“Wesley Sneijder has joined French Ligue 1 side”*, *“Wesley Sneijder has joined Nice”*, and *“Wesley Sneijder has been on a free transfer”*. However, since SRL frames are centered around verbs, it can only extract two STUs: *“Netherlands midfielder Wesley Sneijder joined French Ligue 1 side Nice”* and *“Netherlands midfielder Wesley Sneijder joined on a free transfer”*. On average, human labels 4.8 SCUs per PyrXSum summary, however, the number is only 2.8 for STUs. Hence, a better semantic unit decomposer needs to be designed to improve Lite³Pyramid’s accuracy.

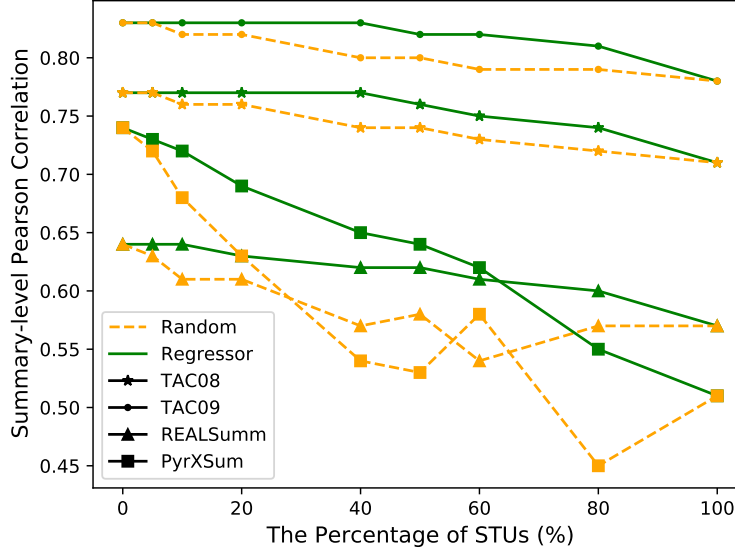


Figure 4.2: Lite^{2.x}Pyramid curves and its comparison to replacing *random* sentences’ SCUs with STUs.

Lastly, **Lite^{2.x}Pyramid** alleviates the problem mentioned above by deferring complex sentences to humans to annotate SCUs for. As shown in Table 4.1, Lite^{2.5}Pyramid, which saves half human effort by substituting 50% sentences’ SCUs with STUs, always has correlation reduction less than half of the difference between Lite²Pyramid and Lite³Pyramid and sometimes even has better system-level correlations than Lite²Pyramid. The full Lite^{2.x}Pyramid curves are shown in Figure 4.2, where the x-axis is the percentage of STUs (the higher means the fewer human efforts involved) and the y-axis is the summary-level Pearson correlation (Figure 4.3 shows system-level correlations). We can see that our Lite^{2.x}Pyramid offers a smoothing transition from semi-automatic Lite²Pyramid to automatic Lite³Pyramid. More importantly, compared to randomly selecting sentences (yellow dash lines), our regressor-based selection achieves a slower correlation reduction, i.e., saving the same amount of human effort our method can retain higher metric quality. Plus, this curve gives people flexible choices per their budget.

The 5-fold (split *by systems*) cross-validation results are in Table 4.2. The same trends are mostly observed. Lite²Pyramid still has 0.06 to 0.21 higher summary-level correlations across all datasets. Lite³Pyramid achieves the best or competitive correlations comparing to other automatic metrics except for the system-level correlations on REALSumm and PyrXSum. And, Lite^{2.x}Pyramid also nicely bridges Lite²Pyramid and Lite³Pyramid and works better than random

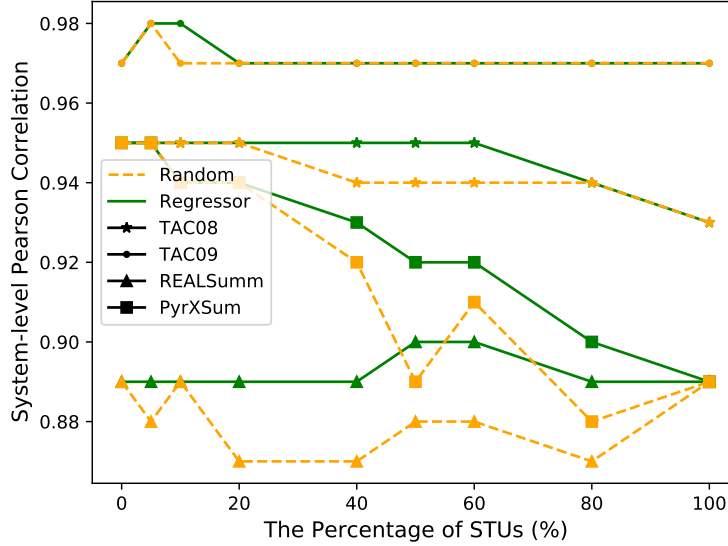


Figure 4.3: Lite^{2.x}Pyramid curves (for system-level correlations) and its comparison to replacing *random* sentences’ SCUs with STUs.

replacement. However, differently, Lite²Pyramid does not get the best system-level correlations on REALSumm and PyrXSum, which may indicate the bigger generalization challenge across different systems.

Takeaway: Lite²Pyramid consistently has the best summary-level correlations and the best system-level correlations in most cases. The automatic Lite³Pyramid also mostly works better than other automatic metrics. Lite^{2.x}Pyramid provides flexible and balanced degrees of automation per budget.

4.5.2 Out-of-the-Box Generalization

We release the finetuned NLI models and the pretrained sentence regressors for future usage, so that they will work as out-of-the-box evaluation metrics for any summarization tasks. Then, a natural question to ask is *how will the metrics perform on a new summarization task?* To better estimate the out-of-the-box performance, we simulate out-of-the-box situations by training the NLI model and the regressor on some dataset(s) and then evaluate metrics on the other dataset(s). For example, in the last big row (starting with TAC08+TAC09+REALSumm) of Table 4.3, we finetune the NLI model and train the regressor on the entire TAC08+TAC09+REALSumm data then evaluate our metrics on PyrXSum only. Meanwhile, we also compare to other metrics. Dif-

Metrics	System-level								Summary-level							
	TAC08		TAC09		REALSumm		PyrXSum		TAC08		TAC09		REALSumm		PyrXSum	
	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ
ROUGE-1	.92	.92	.93	.88	.76	.66	1.0	1.0	.61	.59	.66	.60	.47	.43	.30	.30
ROUGE-2	.96	.96	.97	.93	.82	.82	1.0	1.0	.63	.60	.67	.61	.43	.41	.30	.30
ROUGE-L	.93	.91	.95	.90	.82	.84	1.0	1.0	.56	.53	.62	.57	.40	.36	.29	.29
AutoSummENG	.94	.89	.96	.91	.51	.40	.60	.60	.64	.62	.68	.62	.33	.32	.07	.07
METEOR	.94	.93	.95	.88	.81	.76	.60	.60	.65	.62	.70	.65	<u>.49</u>	.46	.26	.26
BEwT-E	.96	.94	.98	.93	.82	.72	.60	.60	.65	.62	.72	.66	.43	.40	.28	.28
S3 pyr	.95	.93	.95	.89	.81	.78	1.0	1.0	.66	.63	.71	.65	<u>.49</u>	.45	.24	.24
S3 resp	.96	.94	.96	.90	.82	.82	1.0	1.0	.66	.64	.71	.65	.48	.44	.22	.22
PyrEval qual	.91	.88	.91	.84	-	-	-	-	.40	.38	.46	.42	-	-	-	-
PyrEval comp	.90	.87	.93	.80	-	-	-	-	.41	.39	.49	.44	-	-	-	-
BertScore	.91	.89	.98	.89	.69	.68	.60	.60	.61	.58	.67	.62	.43	.40	.12	.12
BertScore (idf)	.93	.90	.97	.89	.70	.68	.60	.60	.61	.58	.68	.63	.44	.41	.10	.10
MoverScore	.95	.92	.96	.90	.47	.46	.20	.20	.64	.61	.71	.65	.37	.34	.14	.14
QAEval EM	.94	.90	.97	.92	.83	.70	.60	.60	.48	.47	.58	.53	.22	.20	.46	.46
QAEval F1	.97	.93	.98	.95	.86	.78	.20	.20	.61	.58	.66	.60	.31	.29	.42	.42
Lite ³ Pyramid	<u>.98</u>	.95	.99	.97	.78	.76	.20	.20	<u>.74</u>	<u>.71</u>	<u>.78</u>	<u>.73</u>	<u>.49</u>	<u>.47</u>	<u>.48*</u>	<u>.48*</u>
Lite ^{2.5} Pyramid	.99	.96	.99	.97	.71	.70	.60	.60	.84	.81	.86	.82	.53	.51	.53*	.53*
Lite ² Pyramid	.99	.98	.99	.98	.74	.72	1.0	1.0	.87	.84	.88	.84	.56	.52	.66*	.66*
Lite ² Pyramid-0	.88	.85	.97	.90	.73	.72	1.0	1.0	.62	.60	.71	.66	.48	.47	.63	.63

Table 4.2: 5-fold (split by systems) cross-validation results. In each column, the **bold** numbers are the best and the underline numbers are the best out of automatic metrics. All Lite²Pyramid-0 numbers are based on $f_{\text{nli}} = l^{3c}$. All other numbers of our metrics are based on $f_{\text{nli}} = p^{2c}$, except that those star* numbers are based on $f_{\text{nli}} = l^{2c}$.

ferent from the numbers in Table 4.1, numbers in Table 4.3 are calculated on the entire meta-evaluation set instead of the average of 5 folds.

It can be observed from Table 4.3 that our Lite²Pyramid retains its advantage in most out-of-the-box situations, especially for summary-level correlation. Though Lite³Pyramid does not always outperform the best metrics, it stays competitive. In addition, Lite^{2.5}Pyramid retains its feature of trading off less than 50% correlation for saving 50% human effort. Surprisingly, learning from more data does not perform better: for PyrXSum, learning from all three other datasets (TAC08+TAC09+REALSumm) gets significantly worse performance than learning from TAC08 only or TAC08+TAC09. We conjecture that the difference between REALSumm (originated from CNN/DM (Hermann et al., 2015)) and PyrXSum (originated from XSum (Narayan et al., 2018a)) leads to a “distribution shift”, which causes the performance drop. Besides, though new metrics have been proposed, ROUGE is still the dominant evaluation metric in the summariza-

		System-level						Summary-level					
		TAC09		REALSumm		PyrXSum		TAC09		REALSumm		PyrXSum	
Metrics		<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ
ROUGE-1		.93	.89	.91	.92	.98	.96	.69	.63	.53	<u>.50</u>	.52	.50
ROUGE-2		.94	.95	.96	<u>.95</u>	<u>.99</u>	.95	.71	.64	.46	.43	.53	.51
ROUGE-L		.96	.92	.94	<u>.95</u>	<u>.99</u>	.95	.66	.59	.46	.42	.52	.51
AutoSummENG		.93	.93	.59	.60	.97	.94	.71	.64	.34	.34	.56	.53
METEOR		.95	.91	.94	.92	<u>.99</u>	.98	.73	.68	<u>.54</u>	.49	<u>.58</u>	<u>.56</u>
BEwT-E		.97	.96	.91	.89	<u>.99</u>	.98	.75	.68	.47	.45	.54	.52
S3 pyr		.97	.92	.96	.94	<u>.99</u>	<u>.99</u>	.75	.67	<u>.54</u>	<u>.50</u>	.57	.54
S3 resp		.96	.94	<u>.97</u>	<u>.95</u>	<u>.99</u>	.98	.74	.68	.52	.48	.57	.54
PyrEval qual		.94	.90	-	-	-	-	.49	.44	-	-	-	-
PyrEval comp		.95	.86	-	-	-	-	.53	.45	-	-	-	-
BertScore		.92	.94	.79	.83	.97	.90	.70	.65	.48	.46	.57	.54
BertScore (idf)		.93	.95	.79	.83	.97	.90	.71	.66	.48	.46	.58	.55
MoverScore		.97	.92	.44	.32	.98	.84	.74	.68	.39	.36	.57	.54
QAEval EM		.88	.94	.88	.86	.95	.95	.64	.55	.28	.27	.29	.27
QAEval F1		.93	.95	.91	.89	.95	.84	.70	.63	.38	.35	.46	.43
TAC08	Lite ³ Pyramid	<u>.99</u>	<u>.97</u>	.92	.93	.97	.90	<u>.78</u>	<u>.72</u>	.53	.48	.56	.53
	Lite ^{2.5} Pyramid	<u>.99</u>	.97	.92	.92	.98	.95	.82	.77	.58	.54	.66	.61
	Lite ² Pyramid	<u>.99</u>	<u>.98</u>	.94	<u>.95</u>	<u>.99</u>	<u>.99</u>	<u>.83</u>	<u>.78</u>	<u>.61</u>	<u>.57</u>	<u>.71</u>	<u>.66</u>
TAC08 +TAC09	Lite ³ Pyramid	-	-	.94	<u>.95</u>	.97	.88	-	-	.52	.49	.56	.53
	Lite ^{2.5} Pyramid	-	-	.93	<u>.95</u>	.97	.96	-	-	.57	.53	.66	.60
	Lite ² Pyramid	-	-	.94	<u>.95</u>	<u>.99</u>	.98	-	-	.59	.56	<u>.71</u>	.65
TAC08 +TAC09 +REALSumm	Lite ³ Pyramid	-	-	-	-	.97	.88	-	-	-	-	.50	.44
	Lite ^{2.5} Pyramid	-	-	-	-	.98	.94	-	-	-	-	.60	.55
	Lite ² Pyramid	-	-	-	-	<u>.99</u>	.94	-	-	-	-	.70	.64

Table 4.3: Out-of-the-box generalization results. In each column, the **bold** numbers are the best and the underline numbers are the best out of automatic metrics.

tion literature. However, based on our comparison, ROUGE is not the best evaluation choice in most cases, while METEOR (Banerjee and Lavie, 2005) and the learning-based metric, S3 (Peyrard et al., 2017), have fairly good correlations with human judgment. Overall, our automatic Lite³Pyramid is on a par with them, having the best performance in 4 cases (4 underline scores in Table 4.3).

Takeaway: When evaluating for a new summarization task with human-labeled SCUs, one could expect that Lite²Pyramid is reliably trustworthy and should be the top choice. Lite³Pyramid is also a fairly good choice for fully automatic evaluation. Finally, our pretrained regressor can guide people on which data examples are more worthy of spending manual effort on annotating SCUs.

Speed: Since SCUs’ collection or STUs’ extraction can be treated as data processing steps, the main speed bottleneck is running the NLI model. When a single TITAN V GPU is available, it takes around 2.5 minutes to evaluate 500 REALSumm (i.e., CNN/DM) examples.

Usage: We provide the support of our metrics through our github repository and we will also incorporate it within the SacreROUGE library (Deutsch and Roth, 2020).

4.5.3 Performance of Individual Modules

NLI. On REALSumm, the finetuned and non-finetuned NLI models get 82.34% and 80.51% accuracy for SCU-presence prediction, respectively. Similarly, 92.53%/87.63% are for TAC08, 93.25%/88.66% are for TAC09, and 92.45%/91.13% are for PyrXSum. Each number is an average of 5 folds (split by examples). As shown in Table 4.1, Lite²Pyramid (with finetuned NLIs) always gets higher correlations than Lite²Pyramid-0 (with non-finetuned NLIs) except for PyrXSum. Therefore, we think NLI accuracy positively affects the results. In our work, we use a RoBERTa (Liu et al., 2019b) based NLI models. Here, to evaluate our metrics’ robustness to different types of NLI models, we test an ALBERT (Lan et al., 2020) based NLI model.⁶ On REALSumm, Lite²Pyramid gets 0.90/0.64 system/summary-level Pearson correlations with human, similar to our RoBERTa-NLI based results (0.89/0.64).

Regressor. On REALSumm, TAC08, TAC09, and XSum, our regressors’ Mean Absolute Errors (MAE) are 0.135, 0.211, 0.206, and 0.090, respectively. On REALSumm, we test a weaker regressor (MAE=0.167), while we get similar results (0.89/0.62 system/summary-level Pearson correlations for Lite^{2.5}Pyramid) to our original regressor (0.90/0.62). However, the sentence selector guided by our regressor always works better than the random selector (shown in Figure 4.2 and Figure 4.3). We think the regressor influences the results by determining the ranking. If we reverse the ranking from the regressor, i.e., replacing SCUs with STUs for more complex sentences, we get lower correlations (0.88/0.60). In our work, we use XGBoost regressor instead of regressors based on pretrained LM because we think to determine the simulation easiness of

⁶ynie/albert-xxlarge-v2-snli_mnli_fever_anli_R1_R2_R3-nli

Model	Fast Abs RL	PtGen	ConvS2S	T-ConvS2S	TransAbs	BertAbs	BertExtAbs	T5	BART	PEGASUS
R2	7.02	9.68	11.58	11.46	10.85	15.63	17.68	21.01	23.96	26.23
Pyramid	0.09	0.09	0.12	0.12	0.07	0.19	0.22	0.29	0.31	0.31

Table 4.4: The ROUGE-2 (R2) and gold Pyramid scores obtained by 10 systems on the 100 XSum testing examples.

sentences, syntactic features are more important than semantic features, and we want to keep the regressor as light-weight as possible. Here, we evaluate a RoBERTa-based regressor on REALSumm and it gets 0.89/0.62 system/summary-level Pearson correlations for Lite^{2.5}Pyramid, which is similar to our XGBoost regressor’s results (0.90/0.62).

4.6 Implementation Details

4.6.1 PyrXSum

Both TAC08/09 (DBL, 2008, 2009) and REALSumm (Bhandari et al., 2020) (examples from CNN/DM (Hermann et al., 2015)) have long and extractive summaries. As a complementary, we collect a new meta-evaluation dataset, PyrXSum, for XSum (Narayan et al., 2018a) which contains short and abstractive summaries. We random sample 100 examples from XSum’s testing set. Then, following Bhandari et al. (2020), we (authors) annotate Semantic Content Units (SCUs) for reference summaries of the 100 examples. After annotation, another non-author native English speaker is invited to double-check the annotated SCUs and give improvement suggestions. Finally, we annotate 2 to 11 SCUs per reference; on average, there are 4.8 SCUs per reference.

Next, we obtain model generated summaries for these 100 examples from 10 abstractive summarization systems: Fast Abs RL (Chen and Bansal, 2018), PtGen (See et al., 2017), ConvS2S and T-ConvS2S (Narayan et al., 2018a), TransAbs and BertAbs and BertExtAbs (Liu and Lapata, 2019), T5 (Raffel et al., 2020), BART (Lewis et al., 2020a), and PEGASUS (Zhang et al., 2020a). We do not include extractive summarization systems because XSum is known to be extremely abstractive and even oracle extractive method has low performance (Narayan et al.,

Instructions (Please read carefully to ensure that your work gets approved as quickly as possible)

Welcome!

We need your help in judging whether some facts are present or not present in a short summary.

For each assignment, you will be prompted with a **short summary** of a news article and a **list of important facts** that are supposed to appear in the summary, each is a short sentence.

Your task is to judge whether each of these facts is *semantically* **Present** or **Not Present** in the summary.

Importantly, **Present** means that **the meaning** of the fact is fully covered by the summary, while **the exact expression/wording can be different**.

An example:

Summary:
bayern munich beat porto 6 - 1 at the allianz arena on tuesday night.

Facts:

1 Bayern Munich beat Porto. *Present*

2 Bayern Munich won 6 - 1. *Present*

3 Bayern Munich won in Champions League. *Not Present*

4 Porto lost the game. *Present*

Summary:

A grand jury in the US state of Texas has decided not to charge anyone over the death of Sandra Bland, who hanged herself in jail.

Facts:

1 A grand jury has decided not to indict anyone in a case. ☐ **Present** ☐ **Not Present**

2 The case involves Sandra Bland. ☐ **Present** ☐ **Not Present**

3 Sandra Bland died in a jail. ☐ **Present** ☐ **Not Present**

4 The jail is at Texas. ☐ **Present** ☐ **Not Present**

5 Sandra Bland died earlier this year. ☐ **Present** ☐ **Not Present**

Figure 4.4: The Amazon Mechanical Turk user interface for collecting human labels of SCUs’ presence.

2018a). For Fast Abs RL, we use their open-source code⁷ to train a model on XSum training set and get its generations for these 100 examples. We directly use the model outputs of PtGen, ConvS2S, and T-ConvS2S, released by Narayan et al. (2018a).⁸ For TransAbs, BertAbs, and BertExtAbs, we also directly use the model outputs released by Liu and Lapata (2019).⁹ For BART (Lewis et al., 2020a) and PEGASUS (Zhang et al., 2020a), we take advantage of the XSum pretrained models released on HuggingFace¹⁰ and generate summaries from them. Lastly, we finetune T5 large on XSum training set via Transformers of HuggingFace (Wolf et al., 2020) and generate summaries from the finetuned model. Table 4.4 lists the ROUGE-2 (R2) (Lin, 2004) results of the 10 systems evaluated only on the 100 examples.

Then, we collect the SCUs’ presence labels for each system summary on Amazon Mechanical Turk. Figure 4.4 illustrates the data annotation instructions and interfaces shown to crowdsourcing workers. The summaries usually only contain one sentence. We estimate it will take around 30-45 seconds for a native English speaker to finish one HIT. Following Bhandari et al. (2020), we pay \$0.15 per HIT, which is respectably higher than the U.S. federal minimum wage requirement. Meanwhile, we select annotators that are located in the U.S., have an approval rate greater than 98%, and have at least 10,000 approved HITs.

We collect 4 responses per summary (100 * 10 * 4 HITs) and finally, 104 workers were involved. After annotation, we filter the annotations from a noisy worker who did 210 HITs but disagreed with the majority in 72% of the time. After this filtering, we obtain an average inter-annotator agreement (Krippendorff’s alpha (Krippendorff, 2011)) of 0.73. Following Bhandari et al. (2020), we use the majority vote to mark the presence of an SCU and break ties by “not present”. Table 4.4 shows the gold Pyramid scores of different systems.

⁷https://github.com/ChenRocks/fast_abs_rl

⁸<https://github.com/EdinburghNLP/XSum>

⁹<https://github.com/nlpyang/PreSumm>

¹⁰<https://huggingface.co/facebook/bart-large-xsum>, <https://huggingface.co/google/pegasus-xsum>

Usually judging the presence of SCUs is considered as a task with little ambiguity, reflected by the high inter-annotator agreements achieved by REAMSumm (0.66) (Bhandari et al., 2020) and our PyrXSum (0.73). To further verify this, on REALSumm, instead of taking the majority vote, we randomly sample 1 out of 4 as the gold label. We conduct this for 3 rounds and test Lite²Pyramid’s correlations with these 3 sets of human labels. We get 0.89/0.63, 0.90/0.63, 0.90/0.63 system/summary-level Pearson correlations, respectively. They are close to each other and also close to the results obtained from the majority vote (0.89/0.64). This means workers give rather consistent SCU-presence labels.

4.6.2 Experimental Details

NLI. For the natural language inference (NLI) used in our work, we take advantage of the pre-trained NLI released by Nie et al. (2020).¹¹ We use the RoBERTa (Liu et al., 2019b) large based version.¹² This model is implemented on HuggingFace’s Transformers (Wolf et al., 2020) and was trained on SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018b), FEVER (Thorne et al., 2018), and ANLI (Nie et al., 2020). We directly use this pretrained model for our Lite²Pyramid-0 metric. When we finetune this model, for simplicity, we always use learning rate=1e-5, linear schedule with warmup, and AdamW (Loshchilov and Hutter, 2018) optimizer, and we always finetune for 2 epochs.

SRL. For Semantic Role Labeling (SRL) model, we use the out-of-the-box SRL model pre-trained by AllenNLP (Gardner et al., 2018).¹³ And it is based the model proposed by Shi and Lin (2019).

¹¹<https://github.com/facebookresearch/anli>

¹²[ymnlp/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli](https://github.com/ymnlp/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli)

¹³<https://demo.allennlp.org/semantic-role-labeling>

Coreference Resolution. For the Coreference Resolution model, we also use the out-of-the-box Coreference model pretrained by AllenNLP (Gardner et al., 2018).¹⁴ And it is based the model proposed by Lee et al. (2018).

Constituency Parsing. For the Constituency Parsing model, we also use the out-of-the-box parser pretrained by AllenNLP (Gardner et al., 2018).¹⁵ And it is based the model proposed by Joshi et al. (2018).

Regressor. The full features we used to train the regressor are: (1) sentence length (in words); (2) linearized parsing tree length (in characters); (3) parsing tree depth; (4) parsing tree depth divided by sentence length; (5) the counts of parsing tree non-terminal tokens.¹⁶ Then, we train the regressor through the XGBoost Python Package¹⁷ and we set the max depth=3, learning rate eta=0.1, number of round=40.

4.7 Conclusion

We propose to combine manual effort and automation for summary evaluation. We introduce a semi-automatic Lite²Pyramid that gains reproducibility by replacing part of human effort with an NLI model. Following it, an automatic Lite³Pyramid is proposed through decomposing references by SRL. Plus, we propose a simple yet effective regressor to decide which sentences are more worthy of labeling SCUs for, leading to flexible transition metrics, Lite^{2..x}Pyramid. Evaluating on four meta-evaluation datasets and comparing to 15 other automatic metrics, Lite²Pyramid consistently has the best summary-level correlations; Lite³Pyramid also performs better or competitively; and Lite^{2..x}Pyramid offers flexible degrees of automation, and its regressor will provide useful or expense-saving guidance for future datasets.

¹⁴<https://demo.allennlp.org/coreference-resolution>

¹⁵<https://demo.allennlp.org/constituency-parsing>

¹⁶WRB, RBR, ADVP, VBG, \$, ", WHADVP, -RRB-, JJR, NAC, PRP, NNS, WP, VBZ, MD, WDT, NP, ADJP, PDT, EX, UH, NN, NFP, SYM, PRP\$, RBS, FRAG, NX, CONJP, RP, WHPP, CC, VBD, LS, ., SBAR, TO, JJ, IN, VP, -LRB-, S, QP, SQ, CD, ", X, POS, XX, PP, PRT, JJS, HYPH, ,, RB, VBN, :, VBP, DT, VB, SINV, UCP, WHNP, NNPS, NNP.

¹⁷<https://xgboost.readthedocs.io/en/latest/python/index.html>

CHAPTER 5: EXTRACTIVE IS NOT FAITHFUL: FAITHFULNESS EVALUATION FOR EXTRACTIVE SUMMARIZATION

5.1 Introduction

Text summarization is the process of distilling the most important information from a source to produce an abridged version for a particular user or task (Maybury, 1999). Although there are many types of text summarization tasks, in this work, we focus on the task of *general purpose single document summarization*. To produce summaries, usually either *extractive summarization* methods, i.e., extracting sentences from the source, or *abstractive summarization* methods, i.e., generating novel text, are applied (Saggion and Poibeau, 2013).

Abstractive summarization attracts more attention from recent works because it can produce more coherent summaries and behaves more like humans (Cohn and Lapata, 2008). Impressive progress has been made for abstractive summarization by large-scale pre-trained models (Lewis et al., 2020b; Zhang et al., 2020a). However, unfaithfulness problems, i.e., hallucinating new information or generating content that contradicts the source, are widely spread across models and tasks (Cao et al., 2018; Maynez et al., 2020). Although these problems do not necessarily get captured by typically-used evaluation metrics, e.g., ROUGE (Lin, 2004), even minor unfaithfulness can be catastrophic and drive users away from real-world applications. Therefore, an increasing volume of research has focused on analyzing (Falke et al., 2019; Maynez et al., 2020; Goyal and Durrett, 2021), evaluating (Kryscinski et al., 2020; Goyal and Durrett, 2021; Wang et al., 2020a; Durmus et al., 2020; Scialom et al., 2021; Xie et al., 2021), or addressing (Cao et al., 2018; Li et al., 2018; Fan et al., 2018b; Chen et al., 2021; Cao and Wang, 2021; Xu et al., 2022; Wan and Bansal, 2022) unfaithfulness problems in abstractive summarization.

Extractive summarization is known to be faster, more interpretable, and more reliable (Chen and Bansal, 2018; Li et al., 2021; Dreyer et al., 2021). And the selection of important information is the first skill that humans learn for summarization (Kintsch and van Dijk, 1978; Brown and Day, 1983). Recently, some works discuss the trade-off between abstractiveness and faithfulness (Ladhak et al., 2022; Dreyer et al., 2021). They find that the more extractive the summary is, the more faithful it is.¹ This may give the community the impression that if the content is extracted from the source, it is guaranteed to be faithful. However, is this always true? In this work, we will show that, unfortunately, it is not.

The problems of extractive summarization are usually referred as *coherence*, *out-of-context*, or *readability* issues (Nanba and Okumura, 2000; Nenkova and McKeown, 2012; Saggion and Poibeau, 2013; Dreyer et al., 2021). Though they may sound irrelevant to faithfulness, some early works give hints of their unfaithful ingredients. Gupta and Lehal (2010) describe the ‘dangling’ anaphora problem – sentences often contain pronouns that lose their referents when extracted out of context, and stitching together extracts may lead to *a misleading interpretation of anaphors*. Barzilay et al. (1999) comment on extractive methods for multi-document summarization, that extracting some similar sentences could produce *a summary biases towards some sources*. Cheung (2008) says that sentence extraction produces extremely incoherent text that *did not seem to convey the gist of the overall controversiality* of the source. These all suggest that even though all information is extracted directly from the source, the summary is not necessarily *faithful* to the source. However, none of these works has proposed an error typology nor quantitatively answered how unfaithful the model extracted summaries are, which motivates us to fill in this missing piece.

In this work, we conduct a thorough investigation of the broad unfaithfulness problems in extractive summarization. Although the literature of abstractive summarization usually limits unfaithful summaries to those that are *not entailed* by the source (Maynez et al., 2020; Kryscinski et al., 2020), we discuss *broadier unfaithfulness* issues including and beyond not-entailment.

¹Note that some previous works seemed to interchange the usage of *factuality* and *faithfulness*. But we think they are slightly different. Thus, we stick to *faithfulness* that represents the property of staying true to the source.

We first design a typology consisting five types of unfaithfulness problems that could happen in extractive summaries: *incorrect coreference*, *incomplete coreference*, *incorrect discourse*, *incomplete discourse*, and *other misleading information* (see definitions in Figure 5.2). Among them, *incorrect coreference* and *incorrect discourse* are not-entailment based errors. An example of incorrect coreference is shown in Summary 1 of Figure 5.1, where *that* in the second sentence should refer to the second document sentence –*But they do leave their trash*, but it incorrectly refers to the first sentence in the summary. Summaries with *incomplete coreferences* or *discourses* are usually entailed by the source, but they can still lead to unfaithful interpretations. Lastly, inspired by *misinformation* (O’Connor and Weatherall, 2019), our misleading information error type refers to other cases where, despite being entailed by the source, the summary still misleads the audience by selecting biased information, giving the readers wrong impressions, etc (see Section 5.2).

We ask humans to label these problems out of 1600 model extracted summaries that are produced by 16 extractive summarization systems for 100 CNN/DM English articles (Hermann et al., 2015). These 16 systems cover both supervised and unsupervised methods, include both recent neural-based and early graph-based models, and extract sentences or elementary discourse units (see Section 5.3). By analyzing human annotations, we find that 30.3% of the 1600 summaries have at least one of the five types of errors. Out of which, 3.9% and 15.4% summaries contain incorrect and incomplete coreferences respectively, 1.1% and 10.7% summaries have incorrect and incomplete discourses respectively, and other 4.9% summaries still mislead the audience without having coreference or discourse issues. The non-negligible error rate demonstrates that extractive is not necessarily faithful. Among the 16 systems, we find that the two oracle extractive systems (that maximize ROUGE (Lin, 2004) against the gold summary by using extracted discourse units or sentences) surprisingly have the most number of problems, while the Lead3 model (the first three sentences of the source document) causes the least number of issues.

We examine whether these problems can be automatically detected by 5 widely-used metrics, including ROUGE (Lin, 2004) and 4 faithfulness evaluation metrics for abstractive summa-

<p>Document: (CNN) <u>Most climbers who try don't succeed in summiting the 29,035-foot-high Mount Everest, the world's tallest peak.</u> <u>But they do leave their trash. Thousands of pounds of it.</u> <u>That's</u> why an experienced climbing group from the Indian army plans to trek up the 8,850-meter mountain to pick up at least 4,000 kilograms (more than 8,000 pounds) of waste from the high-altitude camps, according to India Today. <u>The mountain is part of the Himalaya mountain range on the border between Nepal and the Tibet region.</u> <u>The 34-member team plans to depart for Kathmandu on Saturday and start the ascent in mid-May.</u> <u>The upcoming trip marks the 50th anniversary of the first Indian team to scale Mount Everest [...]</u> <u>More than 200 climbers have died attempting to climb the peak, part of a UNESCO World Heritage Site. The Indian expedition isn't the first attempt to clean up the trash left by generations of hikers[...]</u></p>	
<p>Summary 1 (<i>incorrect coreference</i>): (CNN) Most climbers who try don't succeed in summiting the 29,035-foot-high Mount Everest, the world's tallest peak. That's why an experienced climbing group from the Indian army plans to trek up the 8,850-meter mountain to pick up at least 4,000 kilograms (more than 8,000 pounds) of waste from the high-altitude camps, according to India Today. [...]</p>	
<p>Summary 2 (<i>incomplete coreference & incorrect discourse</i>) : That's why an experienced climbing group from the Indian army plans to trek up the 8,850-meter mountain to pick up at least 4,000 kilograms More than 200 climbers have died to clean up the trash [...]</p>	
<p>Summary 3 (<i>incomplete discourse & incomplete coreference</i>): But they do leave their trash. Thousands of pounds of it. [...]</p>	

Figure 5.1: An example from CNN/DM (Hermann et al., 2015) testing set showing the first four types of unfaithfulness problems defined in section 5.2. The three summaries are generated by NeuSumm (Zhou et al., 2018a) Oracle (disco) (Xu et al., 2020a), and BERT+LSTM+PN+RL (Zhong et al., 2019), respectively. All extracted sentences or discourse units are underlined in the document. The problematic parts are **bolded** in the summary. The incorrect reference in the summary is marked with **red**, and the correct reference is marked with **blue** in the document. We replace non-relevant sentences with [...].

rization (FactCC (Kryscinski et al., 2020), DAE (Goyal and Durrett, 2020), QuestEval (Scialom et al., 2021), BERTScore (Zhang et al., 2020c)). We find that, except BERTScore, they have either no or small correlations with human labels. We design a new metric, ExtEval, for extractive summarization. It contains four sub-metrics that are used to detect incorrect coreference, incomplete coreference, incorrect or incomplete discourse, and sentiment bias, respectively. We show that ExtEval performs best at detecting unfaithful extractive summaries (see Section 5.4 for more details). Finally, we discuss the generalizability and future directions of our work in Section 5.5.

Type	Definition	Rationale
Incorrect Coreference	An anaphor in the summary refers to a different entity from what the same anaphor refers to in the document. The anaphor can be a pronoun (<i>they, she, he, it, this, that, those, these, them, her, him, their, her, his</i> , etc.) or a ‘determiner (<i>the, this, that, these, those, both</i> , etc.) + noun’ phrase.	Not-entailment
Incomplete Coreference	An anaphor in the summary has ambiguous or no antecedent.	Ambiguous interpretation
Incorrect Discourse	A sentence with a discourse linking term (e.g., <i>but, and, also, on one side, meanwhile</i> , etc.) or a discourse unit (usually appears as a sub-sentence) falsely connects to the following or preceding context in the summary, which leads the audience to infer a non-existing fact, relation, etc.	Not-entailment
Incomplete Discourse	A sentence with a discourse linking term or a discourse unit has no necessary following or preceding context to complete the discourse.	Ambiguous interpretation
Other Misleading Information	Other misleading problems include but do not limit to leading the audience to expect a different consequence and conveying a dramatically different sentiment.	Bias and wrong impression

Figure 5.2: Our **typology** of broad unfaithfulness problems in extractive summarization.

In summary, our contributions are (1) a taxonomy of broad unfaithfulness problems in extractive summarization, (2) a human-labeled evaluation set with 1600 examples from 16 diverse extractive systems, (3) meta-evaluations of 5 existing metrics, (4) a new faithfulness metric (ExtEval) for extractive summarization. Overall, we want to remind the community that even when the content is extracted from the source, there is still a chance to be unfaithful. Hence, we should be aware of these problems, be able to detect them, and eventually resolve them to achieve a more reliable summarization.

Github repository: https://github.com/ZhangShiyue/extractive_is_not_faithful

5.2 Broad Unfaithfulness Problems

In this section, we will describe the five types of broad unfaithfulness problems (Figure 5.2) we identified for extractive summarization under our typology. In previous works about abstractive summarization, *unfaithfulness* usually only refers to the summary being *not entailed* by the

source (Maynez et al., 2020; Kryscinski et al., 2020). The formal definition of entailment is t entails h if, typically, a human reading t would infer that h is most likely true (Dagan et al., 2005). While we also consider being *not entailed* as one of the unfaithfulness problems, we will show that there is still a chance to be unfaithful despite being entailed by the source. Hence, we call the five error types we define here the ‘broad’ unfaithfulness problems, and we provide a rationale for each error type in Figure 5.2.

The most frequent unfaithfulness problem of abstractive summarization is the presence of incorrect entities or predicates (Gabriel et al., 2021; Pagnoni et al., 2021), which can never happen within extracted sentences (or elementary discourse units²). For extractive summarization, the problems can only happen ‘across’ sentences (or units).³ Hence, we first define four error types about *coreference* and *discourse*. Following SemEval-2010 (Màrquez et al., 2013), we define coreference as the mention of the same textual references to an object in the discourse model, and we focus primarily on *anaphors* that require finding the correct antecedent. We ground our discourse analysis for systems that extract sentences in the Penn Discourse Treebank (Prasad et al., 2008), which considers the discourse relation between sentences as “lexically grounded”. E.g., the relations can be triggered by subordinating conjunctions (*because, when, etc.*), coordinating conjunctions (*and, but, etc.*), and discourse adverbials (*however, as a result, etc.*). We refer to such words as *discourse linking terms*. For systems that extract discourse units, we follow the Rhetorical Structure Theory (Mann and Thompson, 1988) and assume every unit potentially requires another unit to complete the discourse.

Finally, inspired by the concept of *misinformation* (incorrect or misleading information presented as fact), we define the fifth error type – *misleading information* that captures other misleading problems besides the other four errors. The detailed definitions of the five error types are as follows:

²Elementary Discourse Unit (or EDU) is a concept from the Rhetorical Structure Theory (Mann and Thompson, 1988). Each unit usually appears as a sub-sentence.

³Even though some may argue that extracted sentences should be read independently, in this work, we take them as a whole and follow their original order in the document. We think this is a reasonable assumption and shares the same spirit of previous works that talk about the coherence issue of extractive summaries (Gupta and Lehal, 2010).

Incorrect coreference happens when the same anaphor is referred to different entities given the summary and the document. The anaphor can be a pronoun (*they, she, he, it*, etc.) or a ‘determiner (*the, this, that*, etc.) + noun’ phrase. This error makes the summary not entailed by the source. An example is Summary 1 of Figure 5.1, where the mention *that* refers to the sentence –*But they do leave their trash. Thousands of pounds of it* – in the document but incorrectly refers to *Most climbers who try don’t succeed in summiting the 29,035-foot-high Mount Everest*. Users who only read the summary may think there is some connection between cleaning up trash and the fact that most climbers do not succeed in summiting the Mount Everest.

Incomplete coreference happens when an anaphor in the summary has ambiguous or no antecedent.⁴ Following the formal definition of entailment, these examples are considered to be entailed by the document. Nonetheless, it sometimes can still cause unfaithfulness, as it leads to ‘ambiguous interpretation’. For example, given the source “Jack eats an orange. John eats an apple” and the faithfulness of “He eats an apple” depends entirely on whom “he” is. Figure 5.1 illustrates an example of incomplete coreference, where Summary 2 starts with *that’s why*, but readers of that summary do not know the actual reason. Please refer to Figure 5.4 for another example with a dangling pronoun and ambiguous antecedents.

Incorrect discourse happens when a sentence with a discourse linking term (e.g., *but*, *and*, *also*, etc.)⁵ or a discourse unit falsely connects to the following or preceding context in the summary, which leads the audience to infer a non-existing fact, relation, etc. An example is shown by Summary 2 in Figure 5.1, where *More than 200 climbers have died* falsely connects to *clean up the trash*, which makes readers believe 200 climbers have died because of cleaning up the trash. But in fact, they died attempting to climb the peak. This summary is also clearly not entailed by the source.

⁴Note that sometimes a ‘determiner + noun’ phrase has no antecedent, but it does not affect the understanding of the summary or there is no antecedent of the mention in the document either. In which case, it is *not* an anaphor, and thus we do *not* consider it as an incomplete coreference.

⁵We do not consider implicit (without a linking term) discourse relations between sentences because it hardly appears in our data and will cause a lot of annotation ambiguity.

Incomplete discourse happens when a sentence with a discourse linking term or a discourse unit has no necessary following or preceding context to complete the discourse. Similar to incomplete coreference, summaries with this error are considered entailed, but the broken discourse makes the summary confusing and thus may lead to problematic interpretations. An example is shown in Figure 5.1. Summary 3 starts with *but*, and readers expect to know what leads to this turning, but it is never mentioned. See Figure 5.5 for another example that may leave readers with a wrong impression because of incomplete discourse.

Other misleading information refers to other misleading problems besides the other four error types. It includes but does not limit to leading the audience to expect a different consequence and conveying a dramatically different sentiment. This error is also difficult to capture using the entailment-based definition. Summaries always select partial content from the source, however, sometimes, the selection can mislead or bias the audience. Gentzkow et al. (2015) show that filtering and selection can result in ‘media bias’. We define this error type so that annotators can freely express whether they think the summary has some bias or leaves them with a wrong impression. The summary in Figure 5.6 is labeled as misleading by two annotators because it can mislead the audience to believe that the football players and pro wrestlers won the contest and ate 13 pounds of steak.

Note that we think it is also valid to separate misleading information and incomplete coreference/discourse, as they are *less* severe in unfaithfulness compared to not-entailment-based incorrect coreference/discourse, but we choose to cover all of them under the ‘broad unfaithfulness’ umbrella for completeness.

5.3 Human Evaluation

In this section, we describe how we ask humans to find and annotate the unfaithfulness problems.

5.3.1 Data

We randomly select 100 articles from CNN/DM test set (Hermann et al., 2015) because it is a widely used benchmark for single-document English summarization and extractive methods perform decently well on it. The dataset is distributed under an Apache 2.0 license.⁶ We use 16 extractive systems to produce summaries, i.e., 1600 summaries in total. We retain the order of sentences or units in the document as their order in the summary.

Ten supervised systems: (1) **Oracle** maximizes the ROUGE between the extracted summary and the ground-truth summary; (2) **Oracle (discourse)** (Xu et al., 2020a) extracts discourse units instead of sentences to maximize ROUGE while considering discourse constraints; (3) **RNN Ext RL** (Chen and Bansal, 2018); (4) **BanditSumm** (Dong et al., 2018); (5) **NeuSumm** (Zhou et al., 2018b); (6) **Refresh** (Narayan et al., 2018b); (7) **BERT+LSTM+PN+RL** (Zhong et al., 2019); (8) **MatchSumm** (Zhong et al., 2020); (9) **HeterGraph** (Wang et al., 2020b); (10) **Histruct+** (Ruan et al., 2022). We implement the Oracle system, and we use the open-sourced code of RNN Ext RL⁷ and output of Oracle (discourse)⁸. We get summaries from Histruct+ using their released code and model.⁹ The summaries of other systems are from REALSumm (Bhandari et al., 2020) open-sourced data.¹⁰

Six unsupervised systems: (1) **Lead3** extracts the first three sentences of the document as the summary; (2) **Textrank** (Mihalcea and Tarau, 2004); (3) **Textrank (ST)**: ST stands for Sentence Transformers (Reimers and Gurevych, 2019); (4) **PacSum (tfidf)** and (5) **PacSum (bert)** (Zheng and Lapata, 2019); (6) **MI-unsup** (Padmakumar and He, 2021). We implement

⁶https://huggingface.co/datasets/cnn_dailymail

⁷https://github.com/ChenRocks/fast_abs_rl

⁸<https://github.com/jiacheng-xu/DiscoBERT>

⁹<https://github.com/QianRuan/histruct>

¹⁰<https://github.com/neulab/REALSumm>

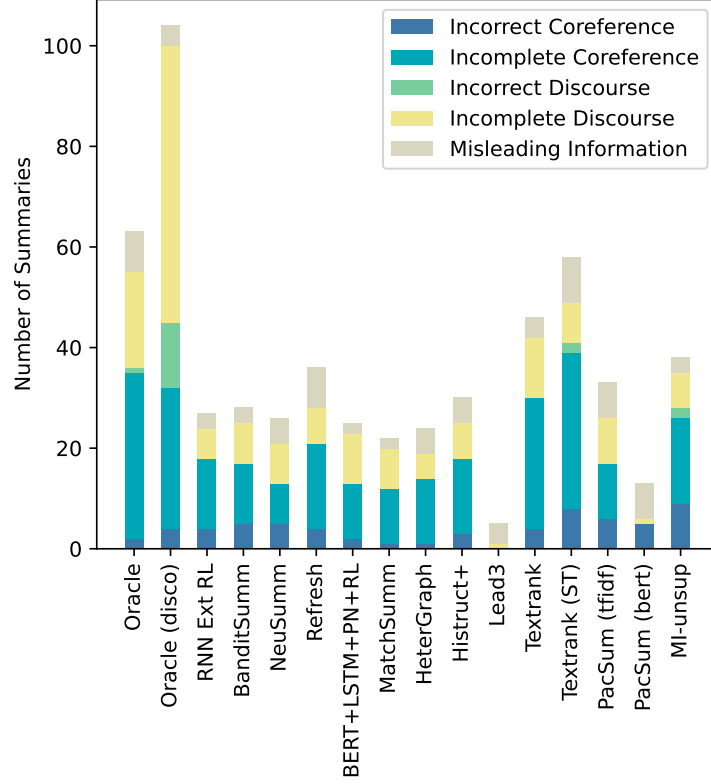


Figure 5.3: The unfaithfulness error distributions of 16 extractive summarization systems.

Lead3 and use the released code of PacSum.¹¹ For Textrank, we use the summa package.¹² For MI-unsup, we directly use the system outputs open-sourced by the authors.¹³

Even though only Oracle (discourse) explicitly uses the discourse structure (the Rhetorical Structure Theory graph), some other systems also implicitly model discourse, e.g., HeterGraph builds a graph of sentences based on word overlap.

5.3.2 Setup

We ask humans to label unfaithfulness problems out of the 1600 system summaries. The annotation interface (HTML page) is shown in Figure 5.8. It first shows the summary and the document. The summary sentences are also underlined in the document. To help with annotation, we

¹¹<https://github.com/mswellhao/PacSum>

¹²<https://github.com/summanlp/textrank>

¹³<https://github.com/vishakhpk/mi-unsup-summ>

run a state-of-the-art coreference resolution model, SpanBERT (Joshi et al., 2020a) via AllenNLP (v2.4.0) (Gardner et al., 2018) on the summary and the document respectively. Then, mentions from the same coreference cluster will be shown in the same color. Since the coreference model can make mistakes, we ask annotators to use them with caution.

Annotators are asked to judge whether the summary has each of the five types of unfaithfulness via five *yes or no* questions and if yes, justify the choice by pointing out the unfaithful parts. Details of the annotation can be found in Section 5.6.1.

Four annotators, two of the authors (PhD students trained in NLP/CL) and two other CS undergraduate students (researchers in NLP/CL), conducted all annotations carefully in about 3 months. Each of the 1600 summaries first was labeled by two annotators independently. Then, they worked together to resolve their differences in annotating incorrect/incomplete coreferences and incorrect/incomplete discourses because these errors have little subjectivity and agreements can be achieved. The judgment of misleading information is more subjective. Hence, each annotator independently double-checked examples that they labeled *no* while their partner labeled *yes*, with their partner’s answers shown to them. They do not have to change their mind if they do not agree with their partner. This step is meant to make sure nothing is missed by accident. In total, 149 examples have at least one misleading label, out of which, 79 examples have both annotators’ misleading labels. In analysis, we only view a summary as misleading when both annotators labeled *yes*, regardless of the fact that they may have different reasons.

5.3.3 Results of Human Evaluation

Finally, we find that 484 out of 1600 (30.3%) summaries contain at least one of the five problems. 63 (3.9%) summaries contain incorrect coreferences, 247 (15.4%) summaries have incomplete coreferences, 18 (1.1%) summaries have incorrect discourses, 171 (10.7%) have incomplete discourses, and 79 (4.9%) summaries are misleading. The error breakdowns for each system are illustrated in Figure 5.3. Note that one summary can have multiple problems, hence why Oracle (discourse) in Figure 5.3 has more than 100 errors.

The nature of different models makes them have different chances to create unfaithfulness problems. For example, the Lead3 system has the least number of problems because the first three sentences of the document usually have an intact discourse, except in a few cases it requires one more sentence to complete the discourse. In contrast, the two Oracle systems have the most problems. The Oracle model often extracts sentences from the middle part of the document, i.e., having a higher chance to cause dangling anaphora or discourse linking. The Oracle (discourse) model contains the most number of incorrect discourses because concatenating element discourse units together increases the risk of misleading context. Furthermore, better systems w.r.t ROUGE scores do not necessarily mean that the summaries are more faithful, e.g., the latest system Histruet+ still contains many unfaithfulness errors, indicating the need to specifically address such faithfulness issues.

Cao et al. (2018) show that about 30% abstractive summaries generated for CNN/DM are not entailed by the source. Also on CNN/DM, FRANK (Pagnoni et al., 2021) finds that about 42% abstractive summaries are unfaithful, including both entity/predicate errors and coreference/discourse/grammar errors. Compared to these findings, extractive summarization apparently has fewer issues. We do note, however, that the quantity is not negligible, i.e., extractive \neq faithful.

5.4 Automatic Evaluation

Here, we analyze whether existing automatic faithfulness evaluation metrics can detect unfaithful extractive summaries. We additionally propose a new evaluation approach, ExtEval.

5.4.1 Meta-evaluation Method

To evaluate automatic faithfulness evaluation metrics (i.e., meta-evaluation) for extractive summarization, we follow the faithfulness evaluation literature of abstractive summarization (Durmus et al., 2020; Wang et al., 2020a; Pagnoni et al., 2021) and compute the correlations between metric scores and human judgment on our meta-evaluation dataset (i.e., the 1600 ex-

amples). Though one summary can have multiple issues for one error type, for simplicity, we use the binary (0 or 1) label as the human judgment of each error type. In addition, we introduce an **Overall** human judgment by taking the *summation* of the five error types. So, the maximum score of Overall is 5. We use Pearson r or Spearman ρ as the correlation measure.

This meta-evaluation method is essentially assessing how well the metric can automatically detect unfaithful summaries, which is practically useful. For example, we can pick out summaries with high unfaithfulness scores and ask human editors to fix them. One underlying assumption is that the metric score is comparable across examples. However, some metrics are example-dependent (one example’s score of 0.5 \neq another example’s score of 0.5), e.g., ROUGE is influenced by summary length (Sun et al., 2019). In practice, we do not observe any significant effect of example dependence on our correlation computation.

To understand the correlation without example-dependence issues, we provide two alternative evaluations *system-level* and *summary-level* correlations, which have been reported in a number of previous works (Peyrard et al., 2017; Bhandari et al., 2020; Deutsch et al., 2021; Zhang and Bansal, 2021). These two correlations assess the effectiveness of the metrics to rank systems.

System-level correlation evaluates *how well the metric can compare different summarization systems*. We denote the correlation measure as K , human scores as h , the metric as m , and generated summaries as s . We assume there are N documents and S systems in the mete-evaluation dataset. The system-level correlation is defined as follows:

$$K_{m,h}^{sys} = K([\frac{1}{N} \sum_{i=1}^N m(s_{i1}), \dots, \frac{1}{N} \sum_{i=1}^N m(s_{iS})], [\frac{1}{N} \sum_{i=1}^N h(s_{i1}), \dots, \frac{1}{N} \sum_{i=1}^N h(s_{iS})])$$

In our case, $N = 100$ and $S = 16$. We use Pearson r or Spearman ρ as the correlation measure K .

Summary-level correlation evaluates *if the metric can reliably compare summaries generated by different systems for the same document*. Using the same notations as above, it is written by:

$$K_{m,h}^{sum} = \frac{1}{N} \sum_{i=1}^N K([m(s_{i1}), \dots, m(s_{iS})], [h(s_{i1}), \dots, h(s_{iS})])$$

5.4.2 Existing Faithfulness Evaluation Metrics

In faithfulness evaluation literature, a number of metrics have been proposed for abstractive summarization. They can be roughly categorized into two groups: entailment classification and question generation/answering (QGQA). Some of them assume that the extractive method is inherently faithful.

We choose FactCC (Kryscinski et al., 2020) and DAE (Goyal and Durrett, 2020) as representative entailment classification metrics. However, since they are designed to check whether each sentence or dependency arc is entailed by the source, we suspect that they cannot detect discourse-level errors. QuestEval (Scialom et al., 2021) is a representative QGQA metric, which theoretically can detect *incorrect coreference* because QG considers the long context of the summary and the document. We also explore BERTScore Precision (Zhang et al., 2020c) that is shown to well correlate with human judgment of faithfulness (Pagnoni et al., 2021; Fischer, 2021), as well as ROUGE-2-F1 (Lin, 2004).

Note that for all metrics except for DAE, we **negate** their scores before computing human-metric correlations because we want them to have higher scores when the summary is more unfaithful, just like our human labels. Table 5.1 shows their original scores for the 16 systems.

	ROUGE-2-F1	FactCC	DAE↓	QuestEval	BERTScore Pre.	ExtEval↓	Human Overall↓
Oracle	25.09	0.95	0.02	0.45	0.92	0.98	0.63
Oracle (discourse)	33.38	0.77	0.00	0.55	0.89	1.65	1.04
RNN Ext RL	12.89	0.97	0.00	0.49	0.95	0.59	0.27
BanditSumm	13.48	0.91	0.00	0.48	0.93	0.57	0.28
NeuSumm	13.69	0.90	0.01	0.48	0.91	0.52	0.26
Refresh	12.96	0.93	0.00	0.48	0.92	0.66	0.36
BERT+LSTM+PN+RL	14.34	0.90	0.00	0.48	0.93	0.59	0.25
MatchSumm	15.42	0.99	0.00	0.48	0.94	0.58	0.22
HeterGraph	14.05	1.00	0.00	0.50	0.94	0.53	0.24
Histruct+	14.43	0.99	0.00	0.63	0.94	0.54	0.30
Lead3	13.03	1.00	0.00	0.49	0.95	0.28	0.05
Textrank	11.06	0.96	0.00	0.46	0.93	0.91	0.46
Textrank (ST)	8.92	0.93	0.02	0.44	0.93	1.07	0.58
PacSum (tfidf)	12.89	0.99	0.01	0.49	0.94	0.59	0.33
PacSum (bert)	13.98	1.00	0.00	0.49	0.95	0.31	0.13
MI-unsup	10.62	0.99	0.00	0.46	0.92	1.05	0.38

Table 5.1: All metric scores and the human Overall score for the 16 extractive systems on the 100 CNN/DM testing examples. The score of a system is the average score of 100 examples. ↓ means the scores are the lower the better.

5.4.3 A New Metric: ExtEval

We introduce ExtEval that is designed for detecting unfaithful extractive summaries. Corresponding to the faithfulness problems defined in Section 5.2, ExtEval is composed of four sub-metrics described as follows. We refer the readers to Section 5.6.2 for more details.

IncorCorefEval focuses on detecting *incorrect coreferences*. Taking advantage of the model-predicted coreference clusters by SpanBERT described in Section 5.3.2, we consider the different cluster mapping of the same mention in the document and summary as *incorrect coreference*.

IncomCorefEval detects *incomplete coreferences*. We also make use of the model-predicted coreference clusters. If the first appeared mention in a summary cluster is a pronoun or ‘determiner + noun’ phrase, and it is not the first mention in the corresponding document cluster, then the summary is considered to have an *incomplete coreference*.

IncomDiscoEval is primarily designed to detect *incomplete discourse*. Concretely, we check for sentences with discourse linking terms and incomplete discourse units. We consider the summary to have a problem if a discourse linking term is present but its necessary context (the previous or next sentence) is missing or a discourse unit misses its previous unit in the same sentence.

Metrics	Incor. Coref.		Incom. Coref.		Incor. Disco.		Incom. Disco.		Mislead.		Overall	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
-ROUGE-2-F1	0.05	0.06	0.03	0.08	-0.07	-0.07	-0.14	-0.10	0.03	0.03	-0.04	0.02
-FactCC	-0.04	-0.04	0.05	0.02	0.24	0.17	0.10	0.03	-0.00	0.01	0.11	0.05
DAE	0.01	0.04	0.04	0.08	0.02	0.04	-0.01	0.02	0.06	0.03	0.05	0.07
-QuestEval	0.09	0.12	0.14	0.15	-0.01	0.01	0.05	0.06	0.08	0.09	0.17	0.19
-BERTScore Pre.	0.08	0.09	0.21	0.20	0.18	0.15	0.29	0.25	0.11	0.12	0.37	0.35
IncorCorefEval	0.25	0.25	0.04	0.04	-0.01	-0.01	-0.00	-0.00	0.04	0.04	0.11	0.08
IncomCorefEval	0.11	0.11	0.48	0.48	0.06	0.06	0.16	0.16	0.01	0.01	0.42	0.42
IncomDiscoEval	0.03	0.03	0.11	0.11	0.20	0.20	0.61	0.61	-0.02	-0.02	0.42	0.38
SentiBias	-0.02	-0.03	0.07	0.05	-0.01	-0.00	0.09	0.08	0.14	0.11	0.13	0.11
ExtEval	0.17	0.13	0.37	0.34	0.14	0.11	0.43	0.36	0.04	0.05	0.54	0.46

Table 5.2: Human-metric correlations. The negative sign (-) before metrics means that their scores are negated to retain the feature that the higher the scores are the more unfaithful the summaries are.

It is important to note that the detected errors also include *incorrect discourse*. However, we cannot distinguish between these two errors.

SentiBias evaluates how different the summary sentiment is from the document sentiment. Sentiment bias is easier to be quantified than other misleading problems. We use the RoBERTa-based (Liu et al., 2019b) sentiment analysis model from AllenNLP (Gardner et al., 2018)¹⁴ to get the sentiments of each sentence. We take the average of sentence sentiments as the overall sentiment of the document or the summary. Then, sentiment bias is measured by the absolute difference between summary sentiment and document sentiment.

ExtEval is simply the summation of the above sub-metrics, i.e., **ExtEval = IncorCorefEval + IncomCorefEval + IncomDiscoEval + SentiBias**. Same as human scores, we make IncorCorefEval, IncomCorefEval, and IncomDiscoEval as binary (0 or 1) scores, while SentiBias is a continuous number between 0 and 1. ExtEval corresponds to the Overall human judgment introduced in Section 5.4.1. Note that when one TiTAN V 12G GPU is available, it takes 0.43 seconds per example to compute ExtEval on average.

¹⁴We also test sentiment analysis tools from Stanza (Qi et al., 2020) and Google Cloud API, but they do not work better.

5.4.4 Meta-Evaluation Results

Table 5.2 shows the human-metric correlations. First, out of the five existing metrics, BERTScore in general works best and has small to moderate (Cohen, 1988) correlations with human judgment, FactCC has a small correlation with incorrect discourse, and other metrics have small or no correlations with human labels. Considering the fact that all these five errors can also happen in abstractive summarization, existing faithfulness evaluation metrics apparently leave these errors behind. Second, the four sub-metrics of ExtEval (IncorCorefEval, IncomCorefEval, IncomDiscoEval, and SentiBias) in general demonstrate better performance than other metrics at detecting their corresponding problems. Lastly, our ExtEval has moderate to large (Cohen, 1988) correlations with the Overall judgment, which is greatly better than all other metrics.

Table 5.3 illustrates the system-level and summary-level correlations of different metrics with human judgment. Note that, for both system-level and summary-level correlations, their correlations are computed between two vectors of length 16 (16 systems), whereas the meta-evaluation method we used in the main paper computes the correlations between two vectors of length 1600 (1600 examples). A smaller sample size will cause a larger variance. This is especially true for system-level correlations, because, following the definitions above, the summary-level correlation ($K_{m,h}^{sum}$) averages across N (in our case, N=100) which can help reduce the variance. Nevertheless, as shown in Table 5.3, our ExtEval achieves the best Pearson and Spearman correlations with the Overall human judgment on both the system level and the summary level. It means ExtEval can rank extractive systems well based on how unfaithful they are. The three sub-metrics (IncorCorefEval, IncomCorefEval, and IncomDiscoEval) perform best at judging which system produces more errors of their corresponding error types. But for detecting misleading information, DAE works best. Out of the 5 existing metrics, BERTScore Precision is the best in general, and on the system level, FactCC also works decently well.

We mainly evaluate ExtEval on the dataset we collected because ExtEval is designed for detecting problematic extractive summaries and is not applicable to abstractive summaries. Nonetheless, we find a subset of SummEval (Fabbri et al., 2021) that contains 4 extractive systems. We

System-level Correlations												
Metrics	Incor. Coref.		Incom. Coref.		Incor. Disco.		Incom. Disco.		Mislead.		Overall	
	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ
-ROUGE-2-F1	0.28	0.59	-0.39	0.08	-0.78	-0.01	-0.88	-0.26	0.01	0.12	-0.71	0.14
-FactCC	0.29	0.34	0.44	0.39	0.81	0.51	0.81	0.60	-0.13	-0.22	0.75	0.54
DAE	0.23	0.26	0.66	0.39	0.11	0.41	0.23	0.74	0.64	0.44	0.50	0.58
-QuestEval	0.27	0.35	0.16	0.40	-0.26	0.33	-0.25	0.36	0.18	0.19	-0.06	0.43
-BERTScore Pre.	0.29	0.30	0.50	0.57	0.70	0.58	0.73	0.58	0.09	0.10	0.74	0.68
IncorCorefEval	0.43	0.12	0.32	0.31	-0.03	0.19	-0.16	-0.02	0.25	0.12	0.11	0.22
IncomCorefEval	0.38	0.34	0.96	0.87	0.52	0.72	0.59	0.56	0.20	0.13	0.85	0.85
IncomDiscoEval	0.30	0.46	0.58	0.76	0.96	0.76	0.92	0.71	-0.06	0.10	0.90	0.88
SentiBias	-0.37	-0.48	0.37	0.18	0.57	0.19	0.69	0.32	0.00	0.01	0.56	0.09
ExtEval	0.37	0.33	0.83	0.84	0.83	0.76	0.84	0.67	0.08	0.09	0.96	0.88
Summary-level Correlations												
Metrics	Incor. Coref.		Incom. Coref.		Incor. Disco.		Incom. Disco.		Mislead.		Overall	
	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ	<i>r</i>	ρ
-ROUGE-2-F1	0.09	0.06	-0.05	-0.01	-0.47	-0.28	-0.37	-0.28	-0.00	0.02	-0.22	-0.13
-FactCC	-0.07	-0.07	0.05	0.04	0.46	0.42	0.13	0.10	0.03	0.03	0.12	0.09
DAE	0.03	0.03	0.16	0.23	0.01	0.11	0.00	0.03	0.20	0.17	0.10	0.14
-QuestEval	0.10	0.13	0.17	0.20	-0.13	-0.06	-0.03	-0.02	0.06	0.08	0.08	0.13
-BERTScore Pre.	0.11	0.12	0.24	0.23	0.48	0.37	0.36	0.30	0.10	0.09	0.36	0.32
IncorCorefEval	0.44	0.44	0.07	0.07	-0.07	-0.07	-0.06	-0.06	0.13	0.13	0.13	0.12
IncomCorefEval	0.13	0.13	0.52	0.52	0.09	0.09	0.23	0.23	0.04	0.04	0.43	0.43
IncomDiscoEval	0.06	0.06	0.15	0.15	0.65	0.65	0.67	0.67	-0.04	-0.04	0.43	0.41
SentiBias	-0.06	-0.06	0.07	0.07	-0.01	0.01	0.06	0.07	0.11	0.11	0.09	0.10
ExtEval	0.23	0.16	0.42	0.37	0.36	0.28	0.48	0.37	0.04	0.07	0.52	0.43

Table 5.3: System-level and summary-level correlations. The negative sign (-) before metrics means that their scores are negated to retain the feature that the higher the scores are the unfaithful the summaries are.

use the average of their consistency (=faithfulness) scores annotated by experts as the gold human scores and compute its correlation with ExtEval. We apply two meta-evaluation methods: (1) Method 1, the same meta-evaluation method as Table 5.2, and (2) Method 2, the system-level evaluation, which is also used by Fabbri et al. (2021), though here we only have 4 systems. The results can be found in Table 5.4. As we can observe, under both methods, our ExtEval achieves the best Spearman correlations and competitive Pearson correlations, which demonstrates the good generalizability of ExtEval.

In summary, our ExtEval is better at identifying unfaithful extractive summaries than the 5 existing metrics we compare to. Its four sub-metrics can be used independently to examine the corresponding unfaithfulness problems.

Metrics	Method 1		Method 2	
	r	ρ	r	ρ
FactCC	-0.04	-0.11	0.68	0.40
QuestEval	-0.04	0.02	-0.46	-0.68
BERTScore Pre.	0.13	0.14	-0.30	0.0
-ExtEval	0.10	0.16	0.31	0.60

Table 5.4: Meta-evaluation results on SummEval (Fabbri et al., 2021). Method 1 refers to the meta-evaluation method used in Section 5.4.1, while Method 2 refers to the system-level correlation used by Fabbri et al. (2021). We negate ExtEval to make higher scores mean more faithful.

5.5 Generalizability

One future direction for resolving these unfaithfulness problems is to use the errors automatically detected by ExtEval as hints for humans or programs to fix the summary by doing necessary yet minimal edits. Here we illustrate the possibility for *incorrect coreference*. We manually examined the automatically detected incorrect coreferences by ExtEval. 28 out of 32 detected incorrect coreferences are true incorrect coreferences¹⁵, which we attempt to fix by developing a simple post-edit program, similar to the revision system proposed by Nanba and Okumura (2000). The program replaces the problematic mention in the summary with the first mention in the correct coreference cluster of the document. We manually checked the corrected examples and found that 16 out of 28 were fixed correctly (see an example in Figure 5.7). We leave the improvement and the extension of post-edit systems for future work.

It is worth noting that all of the five error types we define in Section 5.2 can also happen in abstractive summarization, though they are less studied and measured in the literature. To our best knowledge, FRANK (Pagnoni et al., 2021) and SNaC (Goyal et al., 2022) have discussed the coreference and discourse errors in the abstractive summaries. Gabriel et al. (2021) define a sentiment error as an adjective or adverb appearing in the summary that contradicts the source, while our misleading information has a more general definition. We hope that our taxonomy can shed some light for future works to explore the broad unfaithfulness of all summarization methods.

¹⁵It shows that ExtEval has high precision of 87.5%. However, we have 60 human-labeled incorrect coreferences, so the recall is only 46.7% (28 out of 60).

5.6 Implementation Details

5.6.1 Human Evaluation Details

We did not choose to label the data on Amazon Mechanical Turk because we think that understanding the concepts of coreference and discourse requires some background knowledge of linguistics and NLP.

Figure 5.8 shows the annotation interface and an example annotation. We ask the expert annotators to justify when they think there exists an unfaithful problem. Specifically, if they think the summary has *incorrect coreferences*, they need to further specify the sentence indices and the mentions. For example, “s2-he” means “he” in the second summary sentence is problematic. Meanwhile, they need to justify their answer by explaining why “s2-he” is an incorrect coreference. For *incomplete coreference*, annotators also need to specify the sentence indices plus mentions, but no explanation is required because it can always be “the mention has no clear antecedent.” For *incorrect discourse*, they need to specify sentence indices and justify their choice. For *incomplete discourse*, they only need to specify sentence indices. We find that many summaries have multiple incomplete coreference or discourse issues. Annotators need to label all of them, separated by “,” e.g., “s2-he, s3-the man”. Lastly, besides these four errors, if they think the summary can still mislead the audience, we ask them to provide an explanation to support it.

To avoid one issue in the summary being identified as multiple types of errors, we give the following priorities: incorrect coreference = incorrect discourse > incomplete coreference = incomplete discourse > other misleading information. If an issue is labeled as one type, it will not be labeled for other equal- or lower-priority types.

5.6.2 ExtEval Details

For **IncomCorefEval**, the list of pronouns we use includes *they, she, he, it, this, that, those, these, them, her, him, their, her, his*, and the list of determiners includes *the, that, this, these*,

those, both. This list only contains frequent terms that appear in our dataset, which is not exhaustive.

The list of linking terms for **IncomDiscoEval** includes *and, so, still, also, however, but, clearly, meanwhile, not only, not just, on one side, on another, then, moreover*. Similarly, the list is not exhaustive, and we only keep frequent terms.

5.7 Conclusion

We conducted a systematic analysis of broad unfaithfulness problems in extractive summarization. We proposed 5 error types and produced a human-labeled evaluation set of 1600 examples. We found that (i) 30.3% of the summaries have at least one of the 5 issues, (ii) existing metrics correlate poorly with human judgment, and (iii) our new faithfulness evaluation metric ExtEval performs the best at identifying these problems. Through this work, we want to raise the awareness of unfaithfulness issues in extractive summarization and stress that *extractive is not equal to faithful*.

Document:

(CNN) The California Public Utilities Commission on Thursday said it is ordering Pacific Gas & Electric Co. to pay a record \$1.6 billion penalty for unsafe operation of its gas transmission system, including the pipeline rupture that killed eight people in San Bruno in September 2010.

Most of the penalty amounts to forced spending on improving pipeline safety. Of the 1.6 billion, 850 million will go to "gas transmission pipeline safety infrastructure improvements," the commission said. Another \$50 million will go toward "other remedies to enhance pipeline safety," according to the commission. "PG&E failed to uphold the public's trust," commission President Michael Picker said. "The CPUC failed to keep vigilant. Lives were lost. Numerous people were injured. Homes were destroyed.

We must do everything we can to ensure that nothing like this happens again." The company's chief executive officer said in a written statement that PG&E is working to become the safest energy company in the United States.

"Since the 2010 explosion of our natural gas transmission pipeline in San Bruno, we have worked hard to do the right thing for the victims, their families and the community of San Bruno," Tony Earley said. "We are deeply sorry for this tragic event, and we have dedicated ourselves to re-earning the trust of our customers and the communities we serve. The lessons of this tragic event will not be forgotten."

On September 9, 2010, a section of PG&E pipeline exploded in San Bruno, killing eight people and injuring more than 50 others.

The blast destroyed 37 homes. PG&E said it has paid more than \$500 million in claims to the victims and victims' families in San Bruno, which is just south of San Francisco.

The company also said it has already replaced more than 800 miles of pipe, installed new gas leak technology and implemented nine of 12 recommendations from the National Transportation Safety Board. According to its website, PG&E has 5.4 million electric customers and 4.3 million natural gas customers. The Los Angeles Times reported the previous record penalty was a \$146 million penalty against Southern California Edison Company in 2008 for falsifying customer and worker safety data. CNN's Jason Hanna contributed to this report.

Summary (*incomplete coreference*):

(CNN) The California Public Utilities Commission on Thursday said it is ordering Pacific Gas & Electric Co. to pay a record \$1.6 billion penalty for unsafe operation of its gas transmission system, including the pipeline rupture that killed eight people in San Bruno in September 2010. According to **its** website, PG&E has 5.4 million electric customers and 4.3 million natural gas customers.

Figure 5.4: An example from CNN/DM (Hermann et al., 2015) testing set showing an *incomplete coreference* error. The summary is generated by BERT+LSTM+PN+RL (Zhong et al., 2019). All extracted sentences are underlined in the document. The word **its** in the summary is ambiguous. It can refer to PG&E or California Public Utilities Commission. The correct coreference should be PG&E in the document.

Document:

(CNN) It's been a busy few weeks for multiples.

The first set of female quintuplets in the world since 1969 was born in Houston on April 8, and the parents are blogging about their unique experience.

Danielle Busby delivered all five girls at the Woman's Hospital of Texas via C-section at 28 weeks and two days, according to CNN affiliate KPRC. Parents Danielle and Adam and big sister Blayke are now a family of eight.

The babies are named Ava Lane, Hazel Grace, Olivia Marie, Parker Kate and Riley Paige. "We are so thankful and blessed," said Danielle Busby, who had intrauterine insemination to get pregnant.

"I honestly give all the credit to my God. I am so thankful for this wonderful hospital and team of people here. They truly all are amazing." You can learn all about their journey at their blog, "It's a Buzz World."

Early news reports said the Busby girls were the first all-female quintuplets born in the U.S.

But a user alerted CNN to news clippings that show quintuplet girls were born in 1959 to Charles and Cecilia Hannan in San Antonio.

All of the girls died within 24 hours. Like the Busby family, Sharon and Korey Rademacher were hoping for a second child.

When they found out what they were having, they decided to keep it a secret from family and friends. That's why they didn't tell their family the gender of baby No. 2 – or that Sharon was actually expecting not one but two girls, according to CNN affiliate WEAR.

And when everyone arrived at West Florida Hospital in Pensacola, Florida, after Sharon gave birth March 11, they recorded everyone's reactions to meeting twins Mary Ann Grace and Brianna Faith.

The video was uploaded to YouTube on Saturday and has been viewed more than 700,000 times. Could you keep it a secret?

Summary (*incomplete discourse*):

The first set of female quintuplets in the world since 1969

was born in Houston on April 8,

Danielle Busby delivered all five girls at the Woman's Hospital of Texas via C-section at 28 weeks and two days,

the Busby girls were the first all-female quintuplets

Figure 5.5: An example from CNN/DM (Hermann et al., 2015) testing set showing an *incomplete discourse* error. The summary is generated by the Oracle (disco) (Xu et al., 2020a) extractive system. All extracted elementary discourse units are underlined in the document. The last summary sentence missed the "born in the u.s" part which may make people think the Busby girls is the first all-female quintuplets not only in US.

Document:

(CNN) It didn't seem like a fair fight.

On one side were hulking football players and pro wrestlers, competing as teams of two to eat as many pounds of steak as they could, combined, in one hour.

On another was a lone 124-pound mother of four.

And sure enough, in the end, Sunday's contest at Big Texan Steak Ranch in Amarillo, Texas, wasn't even close.

Molly Schuyler scarfed down three 72-ounce steaks, three baked potatoes, three side salads, three rolls and three shrimp cocktails – far outpacing her heftier rivals.

That's more than 13 pounds of steak, not counting the sides.

And she did it all in 20 minutes, setting a record in the process.

"We've been doing this contest since 1960, and in all that time we've never had anybody come in to actually eat that many steaks at one time," Bobby Lee, who co-owns the Big Texan, told CNN affiliate KVII. "So this is a first for us, and after 55 years of it, it's a big deal."

In fairness, Schuyler isn't your typical 124-pound person. The Nebraska native, 35, is a professional on the competitive-eating circuit and once gobbled 363 chicken wings in 30 minutes.

Wearing shades and a black hoodie, Schuyler beat four other teams on Sunday, including pairs of football players and pro wrestlers and two married competitive eaters.

She also broke her own Big Texan record of two 72-ounce steaks and sides, set last year, when she bested previous record-holder Joey "Jaws" Chestnut.

...

Summary (*other misleading information*):

On one side were hulking football players and pro wrestlers, competing as teams of two to eat as many pounds of steak as they could, combined, in one hour.

And sure enough, in the end, Sunday's contest at Big Texan Steak Ranch in Amarillo, Texas, wasn't even close.

That's more than 13 pounds of steak, not counting the sides.

Figure 5.6: An example from CNN/DM (Hermann et al., 2015) testing set showing a *other misleading information* error. The summary is generated by the HeterGraph (Wang et al., 2020b) extractive system. All extracted sentences are underlined in the document. If readers only read the summary, they may think the football players and pro wrestlers won the contest and ate 13 pounds of steak.

Document:

(CNN) North Korea accused Mexico of illegally holding one of its cargo ships Wednesday and demanded the release of the vessel and crew.

The ship, the Mu Du Bong, was detained after it ran aground off the coast of Mexico in July.

Mexico defended the move Wednesday, saying it followed proper protocol because the company that owns the ship, North Korea's Ocean Maritime Management company, has skirted United Nations sanctions.

...

But An Myong Hun, North Korea's deputy ambassador to the United Nations, said there was no reason to hold the Mu Du Bong and accused Mexico of violating the crew members' human rights by keeping them from their families.

"Mu Du Bong is a peaceful, merchant ship and it has not shipped any items prohibited by international laws or regulations," An told reporters at the United Nations headquarters Wednesday. "And we have already paid full compensation to Mexican authorities according to its domestic laws."

According to Mexico's U.N. mission, the 33 North Korean nationals who make up the vessel's crew are free, staying at a hotel in the port city of Tuxpan and regularly visiting the ship to check on it.

They will soon be sent back to North Korea with help from the country's embassy,

Mexican authorities said.

In the case of the Chong Chon Gang, Panamanian authorities found it was carrying undeclared weaponry from Cuba – including MiG fighter jets, anti-aircraft systems and explosives – buried under thousands of bags of sugar.

Panama seized the cargo and held onto the ship and its crew for months. North Korea eventually agreed to pay a fine of \$666,666 for the vessel's release. CNN's Jethro Mullen contributed to this report.

Original Summary (incorrect coreference):

(CNN) North Korea accused Mexico of illegally holding one of its cargo ships Wednesday and demanded the release of the vessel and crew.

The ship, the Mu Du Bong, was detained after it ran aground off the coast of Mexico in July.

They will soon be sent back to North Korea with help from the country's embassy, Mexican authorities said.

Automatically Corrected Summary:

(CNN) North Korea accused Mexico of illegally holding one of its cargo ships Wednesday and demanded the release of the vessel and crew.

The ship, the Mu Du Bong, was detained after it ran aground off the coast of Mexico in July.

the crew members' will soon be sent back to North Korea with help from the country's embassy, Mexican authorities said.

Figure 5.7: An example of post-correction with ExtEval. In the original summary, *they* refers to *the vessel and crew* in the summary, but it only refers to *the crew* in the document. In the corrected summary, the automated program successfully replaces *they* with *the crew members'* though with a minor grammar issue.

Instructions (Please read carefully to ensure that your work gets approved as quickly as possible!)

Welcome!

We need your help in identify unfaithfulness issues in the extracted summary.

These issues can be:

1. Incorrect Conference: An anaphora in the summary refers to a different entity from what the same anaphora refers to in the document. The anaphora can be a pronoun (they, she, he, it, this, that, those, these, them, her, him, their, her, his, etc.) or a determiner (the, this, that, these, those, both, etc.) + noun phrase.

2. Incomplete Conference: An anaphora in the summary has ambiguous or no antecedent.

3. Incorrect Discourse: A sentence with a discourse linking term (e.g., but, and, also, on one side, meanwhile, etc.) or a discourse unit (usually appears as a sub-sentence) falsely connects to the following or preceding context in the summary, which leads the audience to infer a non-existing fact, relation, etc.

4. Incomplete Discourse: A sentence with a discourse linking term or a discourse unit has no necessary following or preceding context to complete the discourse.

5. Other Misleading Information: Misleading problems include but do not limit to leading the audience to expect a different consequence and conveying a dramatically different sentiment.

Please note that:

1. The summary is composed of extracted sentences (or discourse units) from the document. Those extracted sentences (or sentences that contain extracted units) are underlined in the document.

2. The underlined sentences are automatically found, so they may not be aligned with the summary. When they do, use the summary as the groundtruth and manually align it back to the document.

3. To help annotation, the predicted conference clusters from a conference resolution model are labeled in the document or the summary with colors. The same color refers to the same conference cluster.

4. Since the conference resolution model is not 100% correct, some conference mentions are missed by the model and some mentions are incorrectly grouped together. Thus, they only serve as hints for annotation but please do not only rely on them.

5. For Incomplete Discourse, the necessary context to complete the semantics does not have to be the immediate following or preceding sentence or unit. As long as the semantics are roughly maintained, there should be no problem. Also, if a discourse unit is short and does not convey much meaning itself, it can be exempted from labeling as an incomplete discourse.

6. Please do not label one issue in the summary for multiple error types. Please follow these priorities: incorrect conference > incorrect discourse > incomplete conference > incomplete discourse > misleading information. If an issue is labeled as one type, it will not be labeled for other equal- or lower-priority types.

Summary:

1. that 's why an experienced climbing group from the indian army plans to trek up the 8,850 - meter mountain

2. to pick up at least 4,000 kilograms

3. more than 200 climbers have died

4. to clean up the trash

5. left by generations of hikers .

Document:

1. (cnn) [3] most climbers who try do n't succeed in summiting [3] the 29,035 - foot - high mount everest , the world 's tallest peak .

2. but [3] they do [2] leave [1] their trash .

3. thousands of pounds of [1] it .

4. [2] that 's why [4] an experienced climbing group from the indian army plans to [5] trek up [3] the 8,850 - meter mountain to pick up at least 4,000 kilograms (more than 8,000 pounds) of waste from the high - altitude camps . according to [6] india 's [7] [7] [7] .

5. [3] the mountain is part of the himalaya mountain range on the border between nepal and the tibet region .

6. [3] the 24 - member team plans to depart for kathmandu on saturday and start [5] the ascent in mid - may .

7. [3] the upcoming trip marks the 50th anniversary of the first indian team to scale [3] mount everest . "

8. sadly , [3] mount everest is now ... called the world 's highest junkyard . " maj .

9. [3] ramvir singh jamwal , the team leader , told [6] india 's [7] [7] .

10. [4] we will target the mountaineering waste from camp 1 (19,695 feet) to the summit . " said [8] jamwal , who has scaled mount everest twice . "

11. there are old cylinders , tents , tins , packets , equipment and other mountaineering waste .

12. apart from [4] our own haversacks weighing 10 kg each , [4] we intend to bring in another 10 kg each on [3] the trip . "

13. more than 200 climbers have died attempting to climb [3] the peak , part of a unesco world heritage site .

14. [3] the indian expedition is n't the first attempt to clean up the trash left by generations of hikers .

15. among the cleanup efforts is the eco everest expedition , an annual trip launched in 2008 that is all about climbing " in an eco - sensitive manner , " bringing old refuse , in addition to that generated during the trip , down for disposal , according to the asian trekking website .

16. last year , nepalese tourism authorities started to require [9] hikers to carry out an extra 18 pounds of garbage , in addition to [9] their own trash and human waste , according to the new york times .

1. Does the summary have any *incorrect conference* problems?

☐ Yes ☒ No

If yes, please specify the sentence indexes plus the problematic mentions (e.g., s1-he, s2-her)

If yes, please justify your choice.

2. Does the summary have any *incomplete conference* problems?

☒ Yes ☐ No

If yes, please specify the sentence indexes plus the problematic mentions (e.g., s1-he, s2-her)

s1-that

3. Does the summary have any *incorrect discourse* problems?

☒ Yes ☐ No

If yes, please specify the indexes of problematic sentences (e.g., s1, s3):

s4

If yes, please justify your choice.

The summary makes it sound that many have died to clean up the trash.

4. Does the summary have any *incomplete discourse* problems?

☐ Yes ☒ No

If yes, please specify the indexes of problematic sentences (e.g., s1, s3):

5. Does the summary cause any *other misleading information* problems?

☐ Yes ☒ No

If yes, please justify your choice.

Figure 5.8: The interface for human annotation.

108

CHAPTER 6: CHEROKEE-ENGLISH MACHINE TRANSLATION AND BEYOND

6.1 Introduction

The Cherokee people are one of the indigenous peoples of the United States. Before the 1600s, they lived in what is now the southeastern United States (Peake Raymond, 2008). Today, there are three federally recognized nations of Cherokee people: the Eastern Band of Cherokee Indians (EBCI), the United Keetoowah Band of Cherokee Indians (UKB), and the Cherokee Nation (CN). The Cherokee language, the language spoken by the Cherokee people, contributed to the survival of the Cherokee people and was historically the basic medium of transmission of arts, literature, traditions, and values (Nation, 2001; Peake Raymond, 2008). However, according to the Tri-Council Res. No. 02-2019, there are only 2,000 fluent first language Cherokee speakers left, and each Cherokee tribe is losing fluent speakers at faster rates than new speakers are developed. UNESCO has identified the dialect of Cherokee in Oklahoma is “definitely endangered”, and the one in North Carolina is “severely endangered”. Language loss is memory loss, identity loss, culture loss, and knowledge loss, and it even affects the health of indigenous people (Whalen et al., 2016). CN started a 10-year language revitalization plan (Nation, 2001) in 2008, and the Tri-Council of Cherokee tribes declared a state of emergency in 2019 to save this dying

Src.	iL RG.ə Dʌ.ə ʋəʏ, ʊəʋəʋ DB RG.ə FT hʔRʊ hʔʏ.
Ref.	They are not of the world, even as I am not of the world.
SMT	It was not the things upon the earth, even as I am not of the world.
NMT	I am not the world, even as I am not of the world.

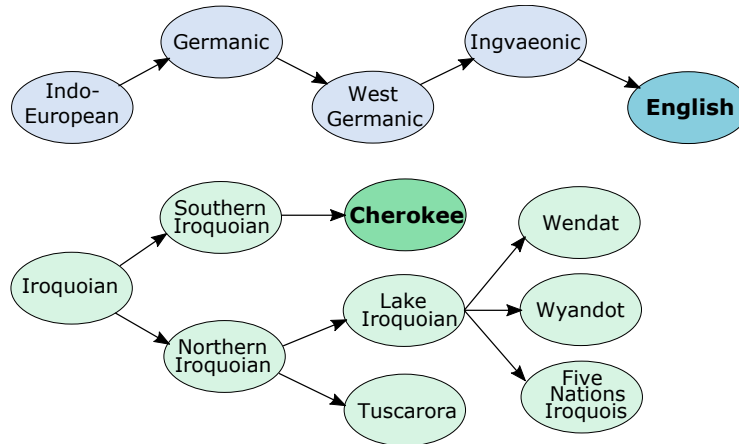
Table 6.1: An example from the development set of ChrEn. NMT denotes our RNN-NMT model.

language. As linguists and NLP researchers, we have the responsibility to address these power imbalances and create a society where space exists for indigenous languages.

To revitalize Cherokee, language immersion programs are provided in elementary schools, and second language programs are offered in universities. However, students have difficulty finding exposure to this language beyond school hours (Albee, 2017). This motivates us to build up English (En) to Cherokee (Chr) machine translation systems so that we could automatically translate or aid human translators to translate English materials to Cherokee. Chr-to-En is also highly meaningful in helping spread Cherokee history and culture.

Therefore, first, we contribute our effort to Cherokee revitalization by constructing a clean **Cherokee-English** parallel dataset, ChrEn, which results in 14,151 pairs of sentences with around 313K English tokens and 206K Cherokee tokens. We also collect 5,210 Cherokee monolingual sentences with 93K Cherokee tokens. Both datasets are derived from bilingual or monolingual materials that are translated or written by first-language Cherokee speakers, then we manually aligned and cleaned the raw data. Our co-author, Prof. Benjamin Frey, is a proficient second-language Cherokee speaker and a citizen of the Eastern Band of Cherokee Indians. Our datasets contain texts of two Cherokee dialects (Oklahoma and North Carolina), and diverse text types (e.g., sacred text, news). To facilitate the development of machine translation systems, we split our parallel data into five subsets: Train/Dev/Test/Out-dev/Out-test, in which Dev/Test and Out-dev/Out-test are for in-domain and out-of-domain evaluation respectively. See an example from ChrEn in Table 6.1 and the detailed dataset description in Section 6.3.1. Recently, we have extended this dataset to containing 17K pairs of sentences. Our data can be found at <https://github.com/ZhangShiyue/ChrEn>.

The translation between Cherokee and English is not easy because the two languages are genealogically disparate. As shown in Figure 6.1, Cherokee is the sole member of the southern branch of the Iroquoian language family and is unintelligible to other Iroquoian languages, while English is from the West Germanic branch of the Indo-European language family. Cherokee uses a unique 85-character syllabary invented by Sequoyah in the early 1820s, which is highly differ-



ent from English’s alphabetic writing system. Cherokee is a polysynthetic language, meaning that words are composed of many morphemes that each have independent meanings. A single Cherokee word can express the meaning of several English words, e.g., ᏊᏵᏩᏍᏔᏅᏏᏉ (widatsinegisi), or *I am going off at a distance to get a liquid object*. Since the semantics are often conveyed by the rich morphology, the word orders of Cherokee sentences are variable. There is no “basic word order” in Cherokee, and most word orders are possible (Montgomery-Anderson, 2008), while English generally follows the Subject-Verb-Object (SVO) word order. Plus, verbs comprise 75% of Cherokee, which is only 25% for English (Feeling, 1975, 1994). See more of Cherokee linguistics in Section 6.2.2.

Hence, to develop translation systems for this low-resource and distant language pair, we investigate various machine translation paradigms and propose phrase-based (Koehn et al., 2003) Statistical Machine Translation (SMT) and RNN-based (Luong et al., 2015) or Transformer-based (Vaswani et al., 2017) Neural Machine Translation (NMT) systems for both Chr-En and En-Chr translations, as important starting points for future works. We apply three semi-supervised methods: using additional monolingual data to train the language model for SMT (Koehn and Knowles, 2017); incorporating BERT (or Multilingual-BERT) (Devlin et al., 2019) representations for NMT (Zhu et al., 2020), where we introduce four different ways to use BERT; and the back-translation method for both SMT and NMT (Bertoldi and Federico, 2009; Lambert et al., 2011; Sennrich et al., 2016b). Moreover, we explore the use of existing X-En parallel datasets of

4 other languages (X = Czech/German/Russian/Chinese) to improve Chr-En/En-Chr performance via transfer learning (Kocmi and Bojar, 2018) or multilingual joint training (Johnson et al., 2017).

Empirically, NMT is better than SMT for in-domain evaluation, while SMT is significantly better under the out-of-domain condition. RNN-NMT consistently performs better than Transformer-NMT. Semi-supervised learning improves supervised baselines in some cases (e.g., back-translation improves out-of-domain Chr-En NMT by 0.9 BLEU). Even though Cherokee is not related to any of the 4 languages (Czech/German/Russian/Chinese) in terms of their language family trees, surprisingly, we find that both transfer learning and multilingual joint training can improve Chr-En/En-Chr performance in most cases. Especially, transferring from Chinese-English achieves the best in-domain Chr-En performance, and joint learning with English-German obtains the best in-domain En-Chr performance. The best results are 15.8/12.7 BLEU for in-domain Chr-En/En-Chr translations; and 6.5/5.0 BLEU for out-of-domain Chr-En/En-Chr translations. Finally, we conduct a 50-example human (expert) evaluation; however, the human judgment does not correlate with BLEU for the En-Chr translation, indicating that BLEU is possibly not very suitable for Cherokee evaluation.

Based on these findings, we develop the first online Cherokee-English machine translation demonstration system: ChrEnTranslate. In addition, our system also supports quality estimation (QE) for both SMT and NMT. QE is an important (missing) component of machine translation systems, which is used to inform users of the reliability of machine-translated content (Specia et al., 2010). Since our models are trained on a very limited number of parallel sentences, it is expected that the translations will be poor in most cases when used by Internet users. Therefore, QE is essential for avoiding misuse and warning users of potential risks. Existing best-performance QE models are usually trained under supervision with quality ratings from professional translators (Fomicheva et al., 2020a). However, we are unable to easily collect a lot of human ratings for Cherokee, due to its state of endangerment. Nonetheless, we test both supervised and unsupervised QE methods: (1) *Supervised*: we use BLEU (Papineni et al., 2002) as the quality rating proxy and train a BLEU regressor; (2) *Unsupervised*: following the uncertain estimation liter-

ature (Lakshminarayanan et al., 2017), we use the ensemble model’s output probability as the estimation of quality. Furthermore, to evaluate how well the QE models perform, we collect 200 human quality ratings (50 ratings for SMT Chr-En, SMT En-Chr, NMT Chr-En, and NMT En-Chr, respectively). We show that our methods obtain moderate to strong correlations with human judgment (Pearson correlation coefficient $\gamma \geq 0.44$).

One main purpose of our system is to allow human-in-the-loop learning. Since limited parallel texts are available, it is important to involve humans, especially experts, in the loop to give feedback and then improve the models accordingly. We develop two different user feedback interfaces for experts and common users, respectively (shown in Figure 6.7). We ask experts to provide quality ratings, correct the model-translated content, and leave open-ended comments; for common users, we allow them to rate how helpful the translation is and to provide open-ended comments. Upon submission, we collected 216 pieces of feedback from 4 experts. We find that experts favor NMT more than SMT because SMT excessively copies from source sentences; according to their ratings and comments, current translation systems *can translate fragments of the source sentence but make major mistakes*. Our naive human-in-the-loop learning, by adding these 216 expert-corrected parallel texts back to the training set, obtains equal or slightly better translation results. Plus, the expert comments shine a light on where the model often makes mistakes. Besides, our demo allows users to input text or choose an example input to translate (shown in Figure 6.6). These examples are from our monolingual databases so that experts will annotate them by providing translation corrections. Finally, to support an intermediate interpretation of the model translations, we visualize the word alignment learned by the translation model and link to cherokeedictionary to provide relevant terms from the dictionary.

Our code is hosted at <https://github.com/ZhangShiyue/ChrEnTranslate> and our on-line website is at <https://chren.cs.unc.edu>. Common users need to accept agreement terms before using our service to avoid misuse; access the expert page <https://chren.cs.unc.edu/expert> requires authorization. We encourage fluent Cherokee speakers to contact us and contribute to our human-in-the-loop learning procedure. A demonstration video of our website is

at <https://youtu.be/-0K8xynDfuE>. In summary, our demo is featured by (1) offering the first online machine translation system for translation between Cherokee and English, which can assist both professional translators and Cherokee learners; (2) documenting human feedback, which, in the long run, expands Cherokee data corpus and allows human-in-the-loop model development.

We then “zoom out” from machine translation and address three important steps on the roadmap of NLP for language revitalization: starting from “before NLP” to “NLP for language education” to “language-specific NLP research”. Before diving into NLP research, we first suggest that NLP practitioners, who are often “outsiders” of indigenous communities, become aware of three important principles: *understand and respect first*, *decolonize research*, and *build a community*. We especially want to promote *building a community*. Since few people are speaking, learning, or studying an endangered language, the knowledge of each individual, the collected resources, and the developed models should be shared as widely and sustainably as possible. Hence, we need a community to support this (see Section 6.5.1). Moreover, language revitalization is an attempt to reverse the decline of a language (Tsunoda, 2013). Fundamentally, this requires an increase in the number of active speakers to bring the language back to day-to-day use (Austin and Sallabank, 2011). Due to the lack of inter-generation transmission, language education in school or online is important. We introduce three approaches for applying NLP techniques in assisting language education (Section 6.5.2): *automated quiz generation*, *automated assessment*, and *community-based language learning*. The last approach connects to our previous point about building a community. Lastly, based on conversations with some Cherokee speakers and researchers, we dive deep into several NLP tools that seem advantageous for community members and may be able to create new usage domains for the language, and we point out the key challenges of their development (Section 6.5.3). Our data and code are available at <https://github.com/ZhangShiyue/RevitalizeCherokee>.

Last but not least, the authors of this line of work come from both the Cherokee community (Benjamin E. Frey) and the NLP community (Shiyue Zhang and Mohit Bansal). Prof. Benjamin

E. Frey is a proficient second-language Cherokee speaker and a citizen of the Eastern Band of Cherokee Indians. He has been teaching Cherokee and contributing to Cherokee revitalization for more than 10 years. He initiated our collaboration and continues bridging the gap between the Cherokee language and language technologies. In addition, we have been talking with some other Cherokee community members, including David Montgomery and Eva Marie Garrouette. Prof. Eva Marie Garrouette from Boston College said: “As a citizen of the Cherokee Nation, I am very concerned for the preservation of my tribe’s endangered language and I am convinced that Dr. Frey’s work represents the most promising project known to me for advancing this goal.” Though by no means the views of our work can represent the whole Cherokee community, our proposals are strongly initiated and motivated by Cherokee community members and grounded by NLP practitioners, and we hope that our work can increase awareness of Cherokee and encourage more work on minority languages

6.2 The Cherokee Language

Starting from this section, we illustrate the situation of endangered languages through the example of Cherokee. We first review its history and linguistics. In the NLP area, we hardly get to know the languages and often let the model learn statistical patterns automatically from the data. However, it is critical to have basic knowledge of the language when contributing to its revitalization.

6.2.1 History of the Cherokee People and Their Language

Tribal Sovereignty. Before encountering Europeans, American Indians were already governing themselves. By drafting treaties with indigenous nations, the colonial powers implicitly recognized their sovereignty. Those treaties are still valid today, and tribal peoples are very much operating as sovereign nations, separate from the US (NCAI, 2020). There are three federally recognized nations of Cherokee people: Cherokee Nation of Oklahoma (CN), United Keetoowah

Band of Cherokee Indians (UKB), and Eastern Band of Cherokee Indians (EBCI). Traditional Cherokee homeland covered parts of what are now eight US states.¹ EBCI is composed of those Cherokees who were able to remain in their homeland. CN is largely comprised of the descendants of those who were forcibly removed to Indian Territory along the infamous Trail of Tears in 1838 (Perdue and Green, 2007), while the UKB is composed largely of those whose ancestors chose to remove themselves west of the Mississippi. Although the three nations are politically independent, they all descend from the same Cherokee people, and maintain common interests, cultural elements, and language.

The Language and its Dialects. Cherokee is the only surviving member of the Southern Iroquoian language family, which have separated from the Northern Iroquoian languages about 4,000 years ago (Julian, 2010). James Mooney identified three main dialects of Cherokee: the Overhill dialect, the Underhill dialect (has died out), and the Middle, or Kituwah dialect. The Overhill dialect is primarily spoken in Oklahoma, and the Middle dialect is predominantly spoken in North Carolina today. Although according to UNESCO, both dialects are endangered, Cherokee is comparatively well-reported among American Indian languages. This is partially due to its writing system known as the 85-character Cherokee syllabary. It was invented in the early 1820s by Sequoyah (Britannica, 2021). The Cherokees have a newspaper written in their own language: the Cherokee Phoenix. The Phoenix, alongside the Cherokee New Testament, formed cornerstones of the Cherokee language in the 1800s on which many current language preservation and archiving projects rest.

Language Endangerment. Cherokee was robustly spoken until around the 1930s. The primary factor being responsible is the US government's "civilization" policy, which aimed to remove American Indians' cultural distinctions (Spring, 2016). Federal boarding schools were created on the model of military institutions by Richard H. Pratt under the philosophy of "kill the Indian, save the man" (Pratt, 2013). American Indian children were sent to residential schools to be educated in how to live in ways more similar to their white contemporaries. School overseers cut

¹North Carolina, South Carolina, Georgia, Kentucky, Tennessee, Alabama, Virginia, and West Virginia.

their hair, forced them to abandon their traditional dress, and punished them for speaking their traditional languages. Beyond the trauma, when they returned to communities, banks, post offices, factories, and grocery stores were all controlled non-locally. People working in them either no longer spoke Cherokee because they were not from Cherokee communities or because their employers were not Cherokee speakers. This transition contributed to the decline of the language in daily use, until the first generation grew up with only English as the language of the home around 1950s (Gulick, 1958; Frey, 2013). Recently, the larger project of language revitalization, of which this paper is a part, endeavors to return the language to regular day-to-day use in the Cherokee communities.

6.2.2 Cherokee Linguistics

Polysynthetic. Cherokee, like most American Indian languages, is polysynthetic. This means that words are primarily composed of a root whose meaning is modified by multiple prefixes and suffixes. The word ᏊᏉ, *gega*, can be divided up: *g-*, *-e-*, *-ga*. The *g-* prefix indicates that the subject of the verb is 1st person singular while the *-ga* suffix indicates that the action happens in the present tense and the aspect is progressive. The verb root *-e-* conveys the idea of motion. The simplest verb form in Cherokee will contain at minimum a root, a pronominal prefix, and a tense/aspect suffix. One oft-noted aspect of Cherokee grammar is its classificatory system, wherein verbs with direct objects must conjugate to indicate the physical shape of the direct object. The verb “I have,” for instance, could appear in any of the following ways: *Agiha* (I have (solid)), *Agineha* (I have (liquid)), *Agwvya* (I have (long & rigid)), *Agina’a* (I have (flexible)), *Agikaha* (I have (animate)). Cherokee also has pre-pronominal prefixes that can specify the geographical location of particular events, such as *wi-* (translocative), which indicates that the action will happen at a distance away from the speaker, and *di-* (cislocative), which indicates the action will happen at a distance approaching the speaker.

Word Order. Word order in Cherokee is dependent on the larger pragmatic context in which the sentence appears, with new information or timeframes occurring before the verb and old or estab-

lished information occurring post-verbally. Subject-object agreement is handled largely via the dual-argument pronominal prefixes. E.g., in “I see it,” ᎠᎾᎾᎾᎾᎾᎾ (*tsigowatiha*), the pronominal prefix *tsi-* indicates 1st person singular (“I”) acting on 3rd person singular (“it”). In ᎠᎾᎾᎾᎾᎾᎾ (*agigowatiha*), we change *tsi-* to *agi-*, which means 3rd person singular acting on 1st person singular.

Person & Number. Although English has only two categories of number: *singular* and *plural*, Cherokee has a third, *dual* category. Therefore, a verb in Cherokee can be conjugated in first, second, or third person and specified for either singular, dual, or plural subjects. Dual and plural prefixes in the first person must then be further subdivided byclusivity, yielding 1st-person dual inclusive (you & I) or exclusive (she/he & I), 1st-person plural inclusive (all of us) or exclusive (they & I). The second person can inflect for dual (you two) or plural (you all). Cherokee does not have a third-person dual form, and speakers usually use the plural form when referring to two third persons.

Verb-centric. Cherokee is very verb-centric, and verbs comprise 75% of Cherokee (Feeling, 1975). Cherokee nouns are divided into root nouns (have no verbal inflection attached to them) and derived nouns (carry verbal morphology). Similarly, Cherokee adjectives can be distinguished from verbs in that their forms cannot carry the tense/aspect morphology typical of actual verbs. Thus, to say someone is skinny, ᎠᎾᎾᎾᎾᎾᎾ (*ulesoda*) carries the pronominal prefix *u-*, indicating 3rd person singular, while ᎠᎾᎾᎾᎾᎾᎾ ᎠᎾᎾᎾᎾᎾᎾ (*ulesoda gesv'i*) marks past tense by adding a separate copula (“to be”) that carries the tense/aspect suffix *-v'i*.

Evidentiality. Cherokee is also marked by a system of evidentiality (indicating whether one has firsthand knowledge of past events, or if one is reporting on hearsay). E.g., one might say ᎠᎾᎾᎾᎾᎾᎾᎾ (*agasgv'i*), “it was raining (and I have firsthand knowledge of this)” vs. ᎠᎾᎾᎾᎾᎾᎾᎾ (*agasge'i*), “it was raining (from what I understand).” Interestingly, this phenomenon applies regardless of the assumed truth of the statement in question.

Phoneme. Cherokee’s phoneme inventory is, like other Iroquoian languages, almost completely bereft of bilabial sounds. It entirely lacks the p or b phonemes, along with f/v , θ/δ , and any r sound. It has six vowels: a , e , i , o , u , and v , and are generally pronounced with continental values, as in Spanish, except for v . Consonant inventory is small, at only 13, and most will be familiar to English speakers. The main exception is the voiceless alveolar fricative ɬ , likely more familiar to Icelandic speakers.

6.3 ChrEn Dataset

6.3.1 Data Collection

It is not easy to collect substantial data for endangered Cherokee. We obtain our data from bilingual or monolingual books and newspaper articles that are translated or written by first-language Cherokee speakers. In the following, we will introduce the data sources and the cleaning procedure and give detailed descriptions of our data statistics.

Parallel Data

Fifty-six percent of our parallel data is derived from the *Cherokee New Testament*. Other texts are novels, children’s books, newspaper articles, etc. These texts vary widely in dates of publication, the oldest being dated to 1860. Additionally, our data encompasses both existing dialects of Cherokee: the Overhill dialect, mostly spoken in Oklahoma (OK), and the Middle dialect, mostly used in North Carolina (NC). These two dialects are mainly phonologically different and only have a few lexical differences (Uchihara, 2016). In this work, we do not explicitly distinguish them during translation. The left pie chart of Figure 6.2 shows the parallel data distributions over text types and dialects, and the complete information is in Table 6.3. Many of these texts were translations of English materials, which means that the Cherokee structures may not be 100% natural in terms of what a speaker might spontaneously produce. But each text was translated by people who speak Cherokee as the first language, which means there is a high probability of

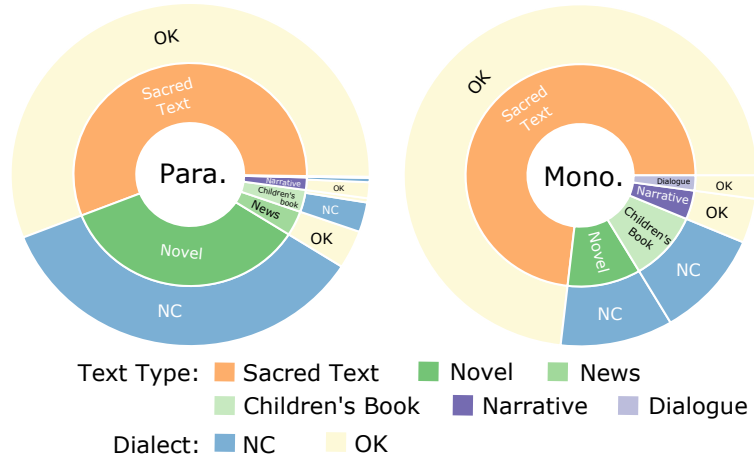


Figure 6.2: The distributions of our parallel (Para.) and monolingual (Mono.) data over text types and dialects.

Statistics	Parallel						Monolingual Total
	Train	Dev	Test	Out-dev	Out-test	Total	
Sentences (or Sentence pairs)	11,639	1,000	1,000	256	256	14,151	5,210
English tokens	257,460	21,686	22,154	5,867	6,020	313,187	-
Unique English tokens	11,606	3,322	3,322	1,605	1,665	13,621	-
% Unseen unique English tokens	-	13.3	13.2	42.1	43.3	-	-
Average English sentence length	22.1	21.7	22.2	22.9	23.5	22.1	-
Cherokee tokens	168,389	14,367	14,373	4,324	4,370	205,823	92,897
Unique Cherokee tokens	32,419	5,182	5,244	1,857	1,881	38,494	19,597
% Unseen unique Cherokee tokens	-	37.7	37.3	67.5	68.0	-	73.7
Average Cherokee sentence length	14.5	14.4	14.3	16.9	17.1	14.5	17.8

Table 6.2: The key statistics of our parallel and monolingual data. Note that “% Unseen unique English tokens” is in terms of the Train split, for example, 13.3% of unique English tokens in Dev are unseen in Train.

grammaticality. These data were originally available in PDF version. We apply the Optical Character Recognition (OCR) via Tesseract OCR engine² to extract the Cherokee and English text. Then our co-author, a proficient second-language speaker of Cherokee, manually aligned the sentences and fixed the errors introduced by OCR. This process is time-consuming and took several months.

²<https://github.com/tesseract-ocr/>

The resulting dataset consists of 14,151 sentence pairs. After tokenization,³ there are around 313K English tokens and 206K Cherokee tokens in total with 14K unique English tokens and 38K unique Cherokee tokens. Notably, the Cherokee vocabulary is much larger than English because of its morphological complexity. This casts a big challenge to machine translation systems because a lot of Cherokee tokens are infrequent. To facilitate machine translation system development, we split this data into training, development, and testing sets. As our data stems from limited sources, we find that if we randomly split the data, some phrases/sub-sentences are repeated in training and evaluation sets, so the trained models will overfit to these frequent patterns. Considering that low-resource translation is usually accompanied by out-of-domain generalization in real-world applications, we provide two groups of development/testing sets. We separate all the sentence pairs from newspaper articles, 512 pairs in total, and randomly split them in half as out-of-domain development and testing sets, denoted by **Out-dev** and **Out-test**. The remaining sentence pairs are randomly split into in-domain **Train**, **Dev**, and **Test**. About 13.3% of unique English tokens and 37.7% of unique Cherokee tokens in Dev have not appeared in Train, while the percentages are 42.1% and 67.5% for Out-dev, which shows the difficulty of the out-of-domain generalization. Table 6.2 contains more detailed statistics; notably, the average sentence length of Cherokee is much shorter than English, which demonstrates that the semantics are morphologically conveyed in Cherokee.

Note that Cherokee-English parallel data is also available on OPUS (Tiedemann, 2012), which has 7.9K unique sentence pairs, 99% of which are the *Cherokee New Testament* that are also included in our parallel data, i.e., our data is bigger and has 6K more sentence pairs that are not sacred texts (novels, news, etc.).

³We tokenize both English and Cherokee by Moses tokenizer (Koehn et al., 2007). For Cherokee, it is equivalent to tokenize by whitespace and punctuation, confirmed to be good enough by our Cherokee-speaker coauthor.

Monolingual Data

In addition to the parallel data, we also collect a small amount of Cherokee monolingual data, 5,210 sentences in total. This data is also mostly derived from Cherokee monolingual books.⁴ As depicted by the right pie chart in Figure 6.2, the majority of monolingual data are also sacred text, which is *Cherokee Old Testament*, and it also contains two-dialect Cherokee texts. Complete information is in Table 6.4. Similarly, we applied OCR to extract these texts. However, we only manually corrected the major errors introduced by OCR. Thus our monolingual data is noisy and contains some lexical errors. As shown in Table 6.2, there are around 93K Cherokee tokens in total with 20K unique Cherokee tokens. This monolingual data has a very small overlap with the parallel data; about 72% of the unique Cherokee tokens are unseen in the whole parallel data. Note that most of our monolingual data have English translations, i.e., it could be converted to parallel data. But it requires more effort from Cherokee speakers and will be part of our future work. For now, we show how to effectively use this monolingual data for semi-supervised gains.

6.3.2 Models

In this section, we will introduce our Cherokee-English and English-Cherokee translation systems. Adopting best practices from low-resource machine translation works, we propose both Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) systems, and for NMT, we test both RNN-based and Transformer-based models. We apply three semi-supervised methods: training language model with additional monolingual data for SMT (Koehn and Knowles, 2017), incorporating BERT or Multilingual-BERT representations into NMT (Zhu et al., 2020), and back-translation for both SMT and NMT (Bertoldi and Federico, 2009; Senrich et al., 2016b). Further, we explore transfer learning (Kocmi and Bojar, 2018) from and multilingual joint training (Johnson et al., 2017) with 4 other languages (Czech, German, Russian, and Chinese) for NMT.

⁴We considered parsing Cherokee Wikipedia. But, according to our coauthor, who is a Cherokee speaker, its content is mostly low-quality.

Title	Speaker/Translator	Date	Type	Dialect	Examples
Cherokee New Testament	Elias Boudinot & Samuel Worcester	1860	Sacred Text	OK	7,957
Charlotte’s Web	Myrtle Driver Johnson	2015	Novel	NC	3,029
Thirteen Moons	Myrtle Driver Johnson	2007	Novel	NC	1,927
A Walk in the Woods	Marie Junaluska	2011	Children’s nonfiction	NC	104
Wolf Wears Shoes (from: Cherokee Stories of the Turtle Island Liars’ Club)	Sequoyah Guess	2012	Traditional narrative	OK	97
The Big Journey of Little Fish	Myrtle Driver Johnson & Abel Catolster	2010	Children’s fiction	NC	97
NSU to host 2017 Inter-Tribal Language Summit	David Crawler	2017	News article	OK	69
Bobby the Bluebird - The Blizzard Blunder	Myrtle Driver Johnson	2016	Children’s fiction	NC	66
A Very Windy Day	Myrtle Driver Johnson	2011	Children’s fiction	NC	59
Sequoyah: The Cherokee Man Who Gave His People Writing	Anna Sixkiller Huckaby	2004	Children’s nonfiction	OK	56
Spearfinger	Nannie Taylor	2008	Traditional narrative	NC	50
Tom Belt Meets Horse	Tom Belt	2008	Personal Narrative	OK	45
The Beast	Marie Junaluska	2012	Children’s fiction	NC	45
Jackson waiting for lung, heart transplants	Anna Sixkiller Huckaby	2017	News article	OK	42
Hannah creates competitive softball league	Anna Sixkiller Huckaby	2017	News article	OK	39
CN re-opens Sequoyah’s Cabin Museum	Anna Sixkiller Huckaby	2017	News article	OK	36
Chance finds passion in creating soap	Anna Sixkiller Huckaby	2016	News article	OK	36
Ice passes on loom weaving knowledge	David Crawler	2017	News article	OK	35
Cherokee National Holiday sees first-ever chunky game	Anna Sixkiller Huckaby	2017	News article	OK	34
Gonzales showcases interpretive Cherokee art	David Crawler	2017	News article	OK	33
Eating healthy on a budget	David Crawler	2017	News article	OK	31
Team Josiah fundraises for diabetes awareness	Anna Sixkiller Huckaby	2017	News article	OK	30
Cherokee Gates scholars reflect on program’s influence	Anna Sixkiller Huckaby	2017	News article	OK	28
‘Mankiller’ premieres June 19 at LA Film Festival	Anna Sixkiller Huckaby	2017	News article	OK	26
Hummingbird, Dart named Cherokee National Treasures	Dennis Sixkiller	2017	News article	OK	25
CNF scholarship applications open Nov. 1	Anna Sixkiller Huckaby	2017	News article	OK	22
Chunestudy feels at home as CHC curator	Anna Sixkiller	2016	News article	OK	20
One Time in Chapel Hill...	Tom Belt	2008	Personal Narrative	OK	20
Ball of Fire (From: Cherokee Narratives: A Linguistic Study)	Durbin Feeling	2018	Personal Narrative	OK	20
Cat Meowing (From: Cherokee Narratives: A Linguistic Study)	Durbin Feeling	2018	Personal Narrative	OK	19
Peas –Our Garden, Our Life	Marie Junaluska	2013	Children’s nonfiction	NC	18
Stopping by Woods on a Snowy Evening	Marie Junaluska	2017	Poetry	NC	16
The Invisible Companion Fox (From: Cherokee Narratives: A Linguistic Study)	Durbin Feeling	2018	Personal Narrative	OK	14
Cherokee Speakers Bureau set for April 12	Anna Sixkiller Huckaby	2018	News article	OK	6

Table 6.3: Parallel Data Sources.

Title	Speaker/Translator	Date	Type	Dialect	Examples
Cherokee Old Testament	Samuel Worcester	1860	Sacred Text	OK	3802
Encyclopedia Brown	Marie Junaluska	2016	Novel	NC	537
Charlie Brown Christmas	Wiggins Blackfox	2020	Children’s fiction	NC	146
Interview with Wilbur Sequoyah	Durbin Feeling	2018	Dialogue	OK	96
One Fish Two Fish Red Fish Blue Fish	Marie Junaluska	2019	Children’s Fiction	NC	91
Climbing The Apple Tree	Marie Junaluska	2020	Children’s Nonfiction	NC	59
How Jack Went to Seek His Fortune	Wiggins Blackfox	2019	Children’s Fiction	NC	50
Trick Or Treat Danny	Wiggins Blackfox	2019	Children’s Fiction	NC	49
Kathy’s Change	Myrtle Driver Johnson	2016	Children’s Fiction	NC	45
Crane And Hummingbird Race	Dennis Sixkiller	2007	Traditional Narrative	OK	44
Ten Apples On Top	Myrtle Driver Johnson	2017	Children’s Fiction	NC	37
Transformation	Durbin Feeling	2018	Personal Narrative	OK	35
Halloween	Wiggins Blackfox	2019	Children’s Fiction	NC	26
Throw It Home	Mose Killer	2018	Personal Narrative	OK	21
Little People	Durbin Feeling	2018	Personal Narrative	OK	19
Hunting Dialogue	Durbin Feeling	2018	Dialogue	OK	18
Two Dogs in On	Durbin Feeling	2018	Personal Narrative	OK	18
Reminiscence	Mose Killer	2018	Personal Narrative	OK	17
The Origin of Evil Magic	Homer Snell	2018	Personal Narrative	OK	17
Water Beast	Sam Hair	2018	Personal Narrative	OK	16
Legal Document	John Littlebones	2018	Personal Narrative	OK	14
The Good Samaritan	Samuel Worcester	1860	Sacred Text	OK	12
My Grandma	Wiggins Blackfox	2018	Children’s Nonfiction	NC	9
Rabbit and Buzzard	Charley Campbell	2018	Personal Narrative	OK	7
Hello Beach	Marie Junaluska	2020	Children’s Nonfiction	NC	7
This Is My Little Brother	Marie Junaluska	2017	Children’s Fiction	NC	7
Diary	Author Unknown	2018	Personal Narrative	OK	6
How to Make Chestnut Bread	Annie Jessan	2018	Personal Narrative	OK	5

Table 6.4: Monolingual Data Sources.

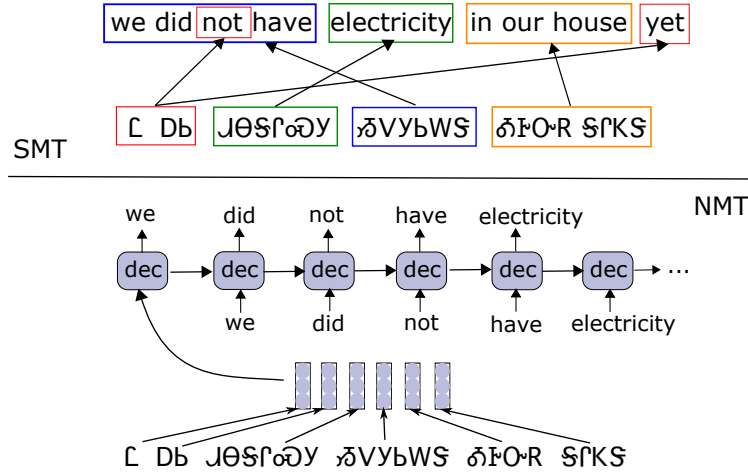


Figure 6.3: A simple illustration of SMT and NMT.

Supervised SMT. SMT was the mainstream of machine translation research before neural models came out. Even if NMT has achieved state-of-the-art performance on many translation tasks, SMT is still very competitive under low-resource and out-of-domain conditions (Koehn and Knowles, 2017). Phrase-based SMT is a dominant paradigm of SMT (Koehn et al., 2003). It first learns a phrase table from the parallel data that translates source phrases to target. Then, a re-ordering model learns to reorder the translated phrases. During decoding, a scoring model scores candidate translations by combining the weights from translation, reordering, and language models, and it is tuned by maximizing the translation performance on the development set. A simple illustration of SMT is shown in Figure 6.3. Note that, as Cherokee and English have different word orders (English follows SVO; Cherokee has variable word orders), one Cherokee phrase could be translated into two English words that are far apart in the sentence. This increases the difficulty of SMT that relies on phrase correspondence and is not good at distant word reordering (Zhang et al., 2017). We implement our SMT systems by Moses (Koehn et al., 2007).

Semi-Supervised SMT. Previous works have shown that SMT can be improved by two semi-supervised methods: (1) A big language model (Koehn and Knowles, 2017), i.e., a language model trained with big target-side monolingual data; (2) Synthesizing bilingual data by back-translating monolingual data (Bertoldi and Federico, 2009; Lambert et al., 2011). Using our Cherokee monolingual data and the publicly available English monolingual data, we test these

two methods. For the first method, we use both parallel and monolingual data to train the language model; for the second method, we back-translate target-language monolingual data into the source language and then combine them with the training set to retrain a source-target SMT model.

Supervised NMT. NMT has mostly dominated recent machine translation research. Especially when a large amount of parallel data is available, NMT surpasses SMT by a large margin; moreover, NMT is good at generating fluent translations because of its auto-regressive generation nature. Koehn and Knowles (2017) pointed out the poor performance of NMT under low-resource and out-of-domain conditions; however, recent work from Sennrich and Zhang (2019) showed that low-resource NMT can be better than SMT by using proper training techniques and hyperparameters. NMT models usually follow encoder-decoder architecture. The encoder encodes the source sentence into hidden representations, then the decoder generates the target sentence word by word by “reading” these representations, as shown in Figure 6.3. We investigate two paradigms of NMT implementations: RNN-based model (Bahdanau et al., 2015) and Transformer-based model (Vaswani et al., 2017). We implement both of them via OpenNMT (Klein et al., 2017). For RNN-NMT, we follow the global attentional model with general attention proposed by Luong et al. (2015). For Transformer-NMT, we mainly follow the architecture proposed by Vaswani et al. (2017) except applying layer normalization before the self-attention and FFN blocks instead of after, which is more robust (Baevski and Auli, 2019).

Semi-Supervised NMT. NMT models can often be improved when more training data is available; therefore, a lot of works have studied semi-supervised approaches that utilize monolingual data to improve translation performance. Similar to SMT, we mainly investigate two semi-supervised methods. The first is to leverage pre-trained language models. Early works proposed shallow or deep fusion methods to rerank NMT outputs or add the language model’s hidden states to NMT decoder (Jean et al., 2015; Gulcehre et al., 2015). Recently, the large-scale pre-trained language model, BERT (Devlin et al., 2019), has achieved impressive success in many NLP tasks. Zhu et al. (2020) showed that incorporating the contextualized BERT representations

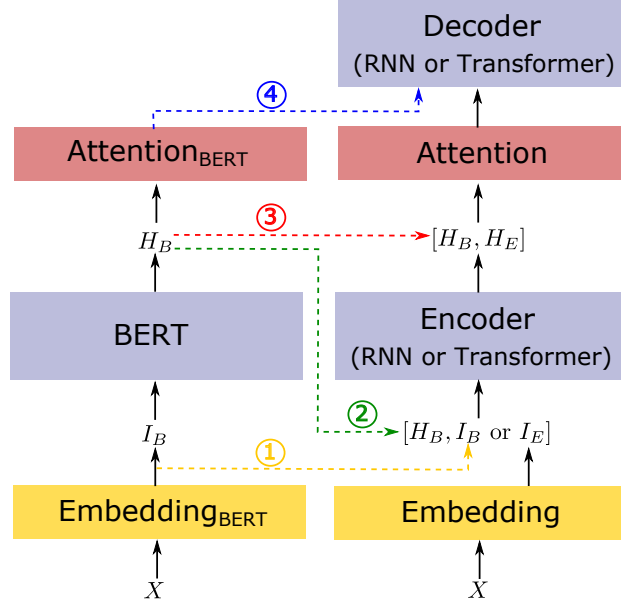


Figure 6.4: The four different ways we proposed to incorporate BERT representations into NMT models.

can significantly improve translation performances. Following but different from this work, we explore four different ways to incorporate BERT representations into NMT models for English-Cherokee translation only.⁵ As depicted in Figure 6.4, we apply BERT representations by: ① Initializing NMT models’ word embedding matrix with BERT’s pre-trained word embedding matrix I_B ; ② Concatenating NMT encoder’s input I_E with BERT’s output H_B ; ③ Concatenating NMT encoder’s output H_E with BERT’s output H_B ; ④ Using another attention to leverage BERT’s output H_B into decoder. Note that ③ and ④ will not be applied simultaneously, and all the combination of these four methods are treated as hyper-parameters, details are in Section 6.3.4. In general, we hope BERT representations can help encoder understand English sentences better and thus improve translation performance. We also test Multilingual-BERT (Devlin et al., 2019) to see if a multilingual pre-trained model can generalize better to a newly encountered language. The second semi-supervised method we try is again the back-translation method. Sennrich et al. (2016b) has shown that applying this method on NMT obtains larger improvement than applying it on SMT, and it works better than the shallow or deep fusion methods.

⁵Because there is no Cherokee BERT. We tried to initialize the decoder embeddings with BERT pre-trained embeddings for Chr-En translation; however, it does not work well.

Transferring & Multilingual NMT. Another important line of research is to improve low-resource translation performance by incorporating knowledge from other language pairs. As mentioned in Section 6.1, Cherokee is the sole member of the southern branch of the Iroquoian language family, so it seems that Cherokee is not “genealogically” related to any high-resource languages in terms of their language family trees. However, it is still interesting to see whether the translation knowledge between other languages and English can help with the translation between Cherokee and English. Hence, in this paper, we will explore two ways of leveraging other language pairs: Transfer learning and Multilingual joint training. Kocmi and Bojar (2018) proposed a simple and effective continual training strategy for the transfer learning of translation models. This method will first train a “parent” model using one language pair until convergence; then continue the training using another language pair, so as to transfer the translation knowledge of the first language pair to the second pair. Johnson et al. (2017) introduced the “many-to-one” and “one-to-many” methods for multilingual joint training of X-En and En-X systems. They achieve this by simply combining training data, except for the “one-to-many” method, every English sentence needs to start with a special token to specify the language to be translated into. We test both the transferring and multilingual methods for Chr-En/En-Chr translations with 4 other X-En/En-X language pairs (X is Czech or German or Russian or Chinese).

6.3.3 Results

Experimental Details

We randomly sample 5K-100K sentences (about 0.5-10 times the size of the parallel training set) from News Crawl 2017⁶ as our English monolingual data. We randomly sample 12K-58K examples (about 1-5 times the size of parallel training set) for each of the 4 language pairs (Czech/German/Russian/Chinese-English) from News Commentary v13 of WMT2018⁷ and

⁶<http://data.statmt.org/news-crawl/en/>

⁷<http://www.statmt.org/wmt18/index.html>

ID	System	Cherokee-English				English-Cherokee			
		Dev	Test	Out-dev	Out-test	Dev	Test	Out-dev	Out-test
S1	SMT	15.0	14.5	6.7	6.4	11.1	9.8	5.4	4.7
S2	+ bigLM	15.3	14.5	6.8	6.5 (± 1.4)	11.3	10.1	5.4	4.7
S3	+ BT	15.4	14.5	6.5	5.9	11.4	9.9	5.7	5.0 (± 1.2)
N4	RNN-NMT	15.7	15.1	2.7	1.8	12.4	11.7	1.1	1.8
N5	+ BERT	-	-	-	-	12.8	12.2	0.7	0.5
N6	+ mBERT	-	-	-	-	12.4	12.0	0.5	0.4
N7	+ BT	16.0	14.9	3.6	2.7	11.4	11.0	1.2	1.5
N8	Transformer-NMT	9.6	9.1	1.1	0.7	7.9	7.4	0.4	0.3
N9	+ BERT	-	-	-	-	8.0	7.2	0.4	0.2
N10	+ mBERT	-	-	-	-	6.8	6.3	0.4	0.2
N11	+ BT	9.9	9.4	1.3	0.5	6.6	5.8	0.4	0.1

Table 6.5: Performance of our supervised/semi-supervised SMT/NMT systems. **Bold** numbers are our best out-of-domain systems together with Table 6.6, selected by performance on Out-dev. ($\pm x$) shows 95% confidence interval.

Bible-uedin (Christodouloupoulos and Steedman, 2015) on OPUS⁸. We apply tokenizer and truecaser from Moses (Koehn et al., 2007). We also apply the BPE tokenization (Sennrich et al., 2016c), but instead of using it as default, we treat it as hyper-parameter. For systems with BERT, we apply the WordPiece tokenizer (Devlin et al., 2019). We compute detokenized and case-sensitive BLEU score (Papineni et al., 2002) using SacreBLEU (Post, 2018).⁹

We implement our SMT systems via Moses (Koehn et al., 2007). **SMT** denotes the base system; **SMT+bigLM** represents the SMT system that uses additional monolingual data to train its language model; SMT with back-translation is denoted by **SMT+BT**. Our NMT systems are implemented by OpenNMT toolkit (Klein et al., 2017). Two baselines are **RNN-NMT** and **Transformer-NMT**. For En-Chr, we also test adding BERT or Multilingual-BERT representations (Devlin et al., 2019), **NMT+BERT** or **NMT+mBERT**, and with back-translation, **NMT+BT**. For Chr-En, we only test **NMT+BT**, treating the English monolingual data size as hyper-parameter. For both En-Chr and Chr-En, we test Transfer learning from and Multilingual joint training with 4 other languages denoted by **NMT+X (T)** and **NMT+X (M)** respectively, where X is Czech/Ger-

⁸<http://opus.nlpl.eu/bible-uedin.php>

⁹BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.4

man/Russian/Chinese. We treat the X-En data size as hyper-parameter. All other detailed model designs and hyper-parameters are introduced in Section 6.3.4.

Quantitative Results

Our main experimental results are shown in Table 6.5 and Table 6.6.¹⁰ Overall, the translation performance is poor compared with the results of some high-resource translations (Sennrich et al., 2016a), which means that current popular SMT and NMT techniques still struggle to translate well between Cherokee and English especially for the out-of-domain generalization.

Chr-En vs. En-Chr. Overall, the Cherokee-English translation gets higher BLEU scores than the English-Cherokee translation. It is reasonable because English has a smaller vocabulary and simpler morphology; thus, it is easier to generate.

SMT vs. NMT. For in-domain evaluation, the best NMT systems surpass SMT for both translation directions. It could result from our extensive architecture hyper-parameter search; or, it supports our conjecture that SMT is not necessarily better than NMT because of the different word orders. But, SMT is dominantly better than NMT for out-of-domain evaluation, which is consistent with the results in Koehn and Knowles (2017).

RNN vs. Transformer. Transformer-NMT performs worse than RNN-NMT, which contradicts the trends of some high-resource translations (Vaswani et al., 2017). We conjecture that Transformer architecture is more complex than RNN and thus requires larger-scale data to train properly. We also notice that Transformer models are very sensitive to hyper-parameters, so it can be possibly improved after a more extensive hyper-parameter search. The best Transformer-NMT has a 5-layer encoder/decoder and 2-head attention, which is smaller-scale than the model used for high-resource translations (Vaswani et al., 2017). Another interesting observation is that previous works have shown applying BPE and using a small vocabulary by setting minimum word frequency are beneficial for low-resource translation (Sennrich et al., 2016c; Sennrich and Zhang,

¹⁰The confidence intervals in Table 6.5 and Table 6.6 are computed by the bootstrap method (Efron and Tibshirani, 1994).

ID	System	Cherokee-English				English-Cherokee			
		Dev	Test	Out-dev	Out-test	Dev	Test	Out-dev	Out-test
N4	RNN-NMT	15.7	15.1	2.7	1.8	12.4	11.7	1.1	1.8
N12	+ Czech (T)	15.8	14.7	2.3	1.8	12.7	12.6	1.8	2.4
N13	+ German (T)	15.9	14.8	2.3	1.1	12.9	12.1	1.8	1.4
N14	+ Russian (T)	16.5	15.8	1.9	1.9	12.6	11.8	1.8	2.3
N15	+ Chinese (T)	16.9	15.8 (± 1.2)	2.0	1.5	12.9	12.9	1.2	0.8
N16	+ Czech (M)	16.6	15.7	2.4	2.0	13.2	12.4	1.1	2.1
N17	+ German (M)	16.6	15.4	2.3	1.4	13.4	12.7 (± 1.0)	0.8	2.0
N18	+ Russian (M)	16.5	15.9	1.9	1.6	13.2	13.1	1.2	1.8
N19	+ Chinese (M)	16.8	16.1	2.2	1.8	13.0	13.0	1.1	1.4

Table 6.6: Performance of our transfer and multilingual learning systems. **Bold** numbers are our best in-domain systems together with Table 6.5, selected by the performance on Dev. ($\pm x$) shows the 95% confidence interval.

2019); however, these techniques are not always being favored during our model selection procedure, as shown in Section 6.3.4.

Supervised vs. Semi-supervised. As shown in Table 6.5, using a big language model and back-translation both only slightly improve SMT baselines on both directions. For English-Cherokee translation, leveraging BERT representations improves RNN-NMT by 0.4/0.5 BLEU points on Dev/Test. Multilingual-BERT does not work better than BERT. Back-translation with our Cherokee monolingual data barely improves performance for both in-domain and out-of-domain evaluations, probably because the monolingual data is also out-of-domain, 72% of the unique Cherokee tokens are unseen in the whole parallel data. For Cherokee-English translation, back-translation improves the out-of-domain evaluation of RNN-NMT by 0.9/0.9 BLEU points on Out-dev/Out-test, while it does not obviously improve in-domain evaluation. A possible reason is that the English monolingual data we used is news data that is not of the same domain as Dev/Test but closer to Out-dev/Out-test so that it helps the model to do domain adaptation. We also investigate the influence of the English monolingual data size. We find that all of the NMT+BT systems perform best when only using 5K English monolingual data, see Figure 6.5 in Section 6.3.4.

Transferring vs. Multilingual. Table 6.6 shows the transfer learning and multilingual joint training results. It can be observed that, in most cases, the in-domain RNN-NMT baseline (N4) can be improved by both methods, which demonstrates that even though the 4 languages are

not related to Cherokee, their translation knowledge can still be helpful. Transferring from the Chinese-English model and joint training with English-German data achieve our best in-domain Cherokee-English and English-Cherokee performance, respectively. However, there is barely an improvement on the out-of-domain evaluation sets, even though the X-En/En-X data is mostly news (same domain as Out-dev/Out-test). On average, multilingual joint training performs slightly better than transfer learning and usually prefers a larger X-En/En-X data size (see details in Section 6.3.4).

Qualitative Results

Automatic metrics are not always ideal for natural language generation (Wieting et al., 2019). As a new language to the NLP community, we are also not sure if BLEU is a good metric for Cherokee evaluation. Therefore, we conduct a small-scale human (expert) pairwise comparison by our coauthor between the translations generated by our NMT and SMT systems. We randomly sample 50 examples from Test or Out-test, anonymously shuffle the translations from two systems, and ask our coauthor to choose which one they think is better.¹¹ As shown in Table 6.7, human preference does not always follow the trends of BLEU scores. For English-Cherokee translation, though the RNN-NMT+BERT (N5) has a better BLEU score than SMT+BT (S3) (12.2 vs. 9.9), it is liked less by humans (21 vs. 29), indicating that BLEU is possibly not a suitable for Cherokee evaluation. A detailed study is beyond the scope of this paper but is an interesting future work direction.

¹¹The author, who conducted this human study, was not involved in the development of MT systems.

Condition		System IDs	Win	Lose
Chr-En	Test	N7 vs. S3	43	7
	Out-test	N7 vs. S2	16	34
En-Chr	Test	N5 vs. S3	21	29
	Out-test	N7 vs. S3	2	48

Table 6.7: Human comparison between the translations generated from our NMT and SMT systems. If A vs. B, “Win” or “lose” means that the evaluator favors A or B. Systems IDs correspond to the IDs in Table 6.5.

6.3.4 Implementation Details

Data and Preprocessing

For semi-supervised learning, we sample additional English monolingual data from News Crawl 2017.¹² We randomly sample 5K, 10K, 20K, 50K, and 100K sentences, which are about half, equal, double, 5-times, 10-times the size of the parallel training set. For transfer and multi-lingual training experiments, we use 12K, 23K, or 58K X-En (X=Czech/German/Russian/Chinese) parallel examples, which are equal, double, and 5-times the size of Chr-En training set. We sample these examples either only from News Commentary v13 of WMT2018¹³ or from both News Commentary and Bible-uedin (Christodouloupoulos and Steedman, 2015) on OPUS¹⁴, because half of in-domain Chr-En data is the Bible. Whenever we sample from Bible-uedin, we keep the sample size as 6K and sample the rest from News Commentary.

For all the data we used, the same tokenizer and truecaser from Moses (Koehn et al., 2007) are applied. For some NMT systems, we also apply the BPE subword tokenization (Sennrich et al., 2016c) with 20,000 merge operations for Cherokee and English separately. For NMT systems with BERT, we apply the WordPiece tokenizer from BERT (Devlin et al., 2019) for English.

¹²<http://data.statmt.org/news-crawl/en/>

¹³<http://www.statmt.org/wmt18/index.html>

¹⁴<http://opus.nlpl.eu/bible-uedin.php>

Before evaluation, the translation outputs are detokenized and detruccased. We use SacreBLEU (Post, 2018)¹⁵ to compute the BLEU (Papineni et al., 2002) scores of all translation systems.

SMT Systems

We implement SMT systems via Moses (Koehn et al., 2007). We train a 3-gram language model (LM) by KenLM (Heafield et al., 2013) and conduct word alignment by GIZA++ (Och and Ney, 2003). Model weights are tuned on the Dev or Our-dev by MERT (Och, 2003).

NMT Systems

Our NMT systems are all implemented by OpenNMT (Klein et al., 2017). As shown in Table 6.5 and Table 6.6, there are 16 NMT systems in total (N4-N19). For each of these systems, We conduct a limited amount of hyper-parameter grid search on Dev or Out-dev. The search space includes applying BPE or not, minimum word frequency threshold, number of encoder/decoder layers, hidden size, dropout, etc. The detailed hyper-parameter tuning procedure will be discussed in the next subsection. During decoding, all systems use beam search with beam size 5 and replace unknown words with source words that have the highest attention weight.

Hyper-parameters

We observed the NMT models, especially the Transformer-NMT models, are sensitive to hyper-parameters. Thus, we did a limited amount of hyper-parameter grid search when developing NMT models. For building vocabulary, we take BPE (Sennrich et al., 2016c) (use or not) and the minimum word frequency (0, 5, 10) as two hyper-parameters. For the model architecture, we explore different number of encoder/decoder layers (1, 2, 3 for RNN; 4, 5, 6 for Transformer), hidden size (512, 1024), embedding size (equals to hidden size, except 768 for BERT), tied decoder embeddings (Press and Wolf, 2017) (use or not), and number of attention heads (2, 4, 8).

¹⁵BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.4

For training techniques, we tune dropout (0.1, 0.2, 0.3), label smoothing (Szegedy et al., 2016) (0.1, 0.2), average decay (1e-4 or not use), batch type (tokens or sentences), batch size (1000, 4000 for tokens; 32, 64 for sents), and warmup steps (3000, 4000). We take the English monolingual data size (5K, 10K, 20K, 50K, 100K) as hyper-parameter when we do back-translation for Cherokee-English translation. We take the size of Czech/German/Russian/Chinese-English parallel data (12K, 23K, 58K) and whether sampling from Bible-uedin (yes or no) as hyper-parameter when we do transfer or multilingual training. Besides, we take how we incorporate BERT as hyper-parameter, and it is chosen from the following five settings and their combinations:

BERT embedding: Initializing NMT models’ word embedding matrix with BERT’s pre-trained word embedding matrix I_B , corresponding to ① in Figure 6.4;

BERT embedding (fix): The same as “BERT embedding” except we fix the word embedding during training;

BERT input: Concatenate NMT encoder’s input I_E with BERT’s output H_B , corresponding to ② in Figure 6.4;

BERT output: Concatenate NMT encoder’s output H_E with BERT’s output H_B , corresponding to ③ in Figure 6.4;

BERT output (attention): Use another attention to leverage BERT’s output H_B into decoder, corresponding to ④ in Figure 6.4;

“BERT embedding” and “BERT embedding (fix)” will not be applied simultaneously, and “BERT output” and “BERT output (attention)” will not be applied simultaneously. Multilingual-BERT is used in the same ways. At most, there are 576 searches per model, but oftentimes, we did less than that because we early cut off unpromising settings. All hyper-parameters are tuned on Dev or Out-dev for in-domain or out-of-domain evaluation, and the model selection is based on translation accuracy on Dev or Out-dev. Table 6.8, Table 6.9, Table 6.10, Table 6.11, Table 6.12, Table 6.13, and Table 6.14 list the hyper-parameters of all the systems shown in the Table 6.5 and Table 6.6. Since our parallel dataset is small (14K sentence pairs), even the slowest experiment, Transformer-NMT+mBERT, only takes 2 minutes per epoch using one Tesla V100 GPU. We

Hyper-parameter	Dev				Out-dev			
	N4	N7	N8	N11	N4	N7	N8	N11
BPE	yes				-			
word min frequency	10				0		10	
encoder layer	2		5		2		5	
decoder layer	2		5		2		5	
hidden size	1024				512		1024	
embedding size	1024				512		1024	
tied decoder embeddings	yes		-		yes		-	yes
head	-		2		-		2	
dropout	0.3	0.5	0.1		0.3		0.1	
label smoothing	0.2		0.1		0.2		0.1	
average decay	1e-4		-		1e-4			
batch type	tokens	sents	tokens		sents		tokens	
batch size	1000	32	4000		32		4000	
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-		rsqrt		-		rsqrt	
warmup steps	-		4000		-		4000	
early stopping	10							
mono. data size	-	5000	-	5000	-	5000	-	5000

Table 6.8: The hyper-parameter settings of **Supervised and Semi-supervised Cherokee-English NMT systems in Table 6.5**. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

train 100 epochs at most and using early stop when the translation accuracy on Dev or Out-dev does not improve for 10 epochs.

English Monolingual Data Size Influence

In the semi-supervised experiments of Cherokee-English, we investigate the influence of the English monolingual data size. As mentioned above, we use 5K, 10K, 20K, 50K, and 100K English monolingual sentences. Figure 6.5 shows its influence on translation performance. It can be observed that increasing English monolingual data size does not lead to higher performance, especially, all NMT+BT systems achieve the best performance when only use 5K English sentences.

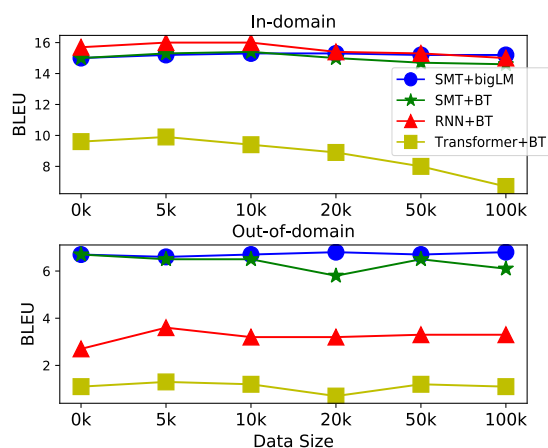


Figure 6.5: The influence of the English monolingual data size on semi-supervised learning performance. The results are on Dev or Out-dev.

Hyper-parameter	Dev				Out-dev			
	N12	N13	N14	N15	N12	N13	N14	N15
BPE	-							
word min frequency	0					5		
encoder layer	2							
decoder layer	2							
hidden size	1024				512			
embedding size	1024				512			
tied decoder embeddings	yes							
head								
dropout	0.3							
label smoothing	0.2							
average decay	1e-4							
batch type	tokens				sents			
batch size	1000				32			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-							
warmup steps	-							
early stopping	10							
X-En data size	11,639			23,278		11,639		23,278
with Bible	no		yes		no	yes		

Table 6.9: The hyper-parameter settings of Transferring Cherokee-English NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

Hyper-parameter	Dev				Out-dev			
	N16	N17	N18	N19	N16	N17	N18	N19
BPE	-							
word min frequency	5					5		
encoder layer	2							
decoder layer	2							
hidden size	1024				512			
embedding size	1024				512			
tied decoder embeddings	yes							
head								
dropout	0.3							
label smoothing	0.2							
average decay	1e-4							
batch type	tokens				sents			
batch size	1000				32			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-							
warmup steps	-							
early stopping	10							
X-En data size	58,195				23,278			
with Bible	yes				no			

Table 6.10: The hyper-parameter settings of Multilingual Cherokee-English NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

Hyper-parameter	Dev							
	N4	N7	N5	N6	N8	N11	N9	N10
BPE	-		-		yes		-	
WordPiece	-		yes		-		yes	
word min frequency	0				5		0	
encoder layer	2				5			
decoder layer	2				5			
hidden size	1024							
embedding size	1024		768		1024		768	
tied decoder embeddings	yes				-			
head	-				2			
dropout	0.5				0.1			
label smoothing	0.2				0.1	0.2	0.1	
average decay	1e-4				-			
batch type	tokens							
batch size	1000				4000			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-				rsqrt			
warmup steps	-				4000			
early stopping	10							
mono. data size	-	5210	-			5210	-	
BERT embedding	-						yes	
BERT embedding (fix)	-		yes	-	-			
BERT input	-		yes	-	-		yes	
BERT output	-		yes	-	-		yes	
BERT output (attention)	-				-			

Table 6.11: The hyper-parameter settings of in-domain Supervised and Semi-supervised English-Cherokee NMT systems in Table 6.5. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

Hyper-parameter	Out-dev							
	N4	N7	N5	N6	N8	N11	N9	N10
BPE	-							
WordPiece	-		yes		-		yes	
word min frequency	10			0	0			
encoder layer	2				5			
decoder layer	2				5			
hidden size	512				1024			
embedding size	512		768		1024		768	
tied decoder embeddings	yes				-	yes	-	
head	-				2			
dropout	0.3	0.5	0.3		0.1			
label smoothing	0.2	0.1	0.2		0.2			
average decay	1e-4				-	1e-4	-	
batch type	sents				tokens			
batch size	32				4000			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-				rsqrt			
warmup steps	-				4000			
early stopping	10							
mono. data size	-	5210	-			5210	-	
BERT embedding	-		yes		-			
BERT embedding (fix)	-						yes	-
BERT input	-		yes		-			
BERT output	-						yes	-
BERT output (attention)	-							

Table 6.12: The hyper-parameter settings of out-of-domain Supervised and Semi-supervised English-Cherokee NMT systems in Table 6.5. Empty fields indicate that hyper-parameter is the same as previous (left) system.

Hyper-parameter	Dev				Out-dev			
	N12	N13	N14	N15	N12	N13	N14	N15
BPE	-							
word min frequency	0				10	5	10	
encoder layer	2							
decoder layer	2							
hidden size	1024				512			
embedding size	1024				512			
tied decoder embeddings	yes							
head								
dropout	0.3							
label smoothing	0.2							
average decay	1e-4							
batch type	tokens				sents			
batch size	1000				32			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-							
warmup steps	-							
early stopping	10							
En-X data size	23,278	11,639	23,278		11,639	23,278		11,639
with Bible	yes	no	yes			no		

Table 6.13: The hyper-parameter settings of Transferring English-Cherokee NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

Hyper-parameter	Dev				Out-dev			
	N16	N17	N18	N19	N16	N17	N18	N19
BPE	-							
word min frequency	5					5		
encoder layer	2							
decoder layer	2							
hidden size	1024				512			
embedding size	1024				512			
tied decoder embeddings	yes							
head								
dropout	0.3							
label smoothing	0.2							
average decay	1e-4							
batch type	tokens				sents			
batch size	1000				32			
optimizer	adam							
learning rate (lr)	5e-4							
lr decay method	-							
warmup steps	-							
early stopping	10							
En-X data size	58,195				23,278			11,639
with Bible	yes				no		yes	no

Table 6.14: The hyper-parameter settings of Multilingual English-Cherokee NMT systems in Table 6.6. Empty fields indicate that hyper-parameter is the same as the previous (left) system.

6.4 ChrEnTranslation System

6.4.1 System Description

Translation Models

As shown in Figure 6.6, our system allows users to choose the statistical or neural model (SMT or NMT).

SMT is more effective for out-of-domain translation between Cherokee and English (Zhang et al., 2020b). We implement phrase-based SMT model via Moses (Koehn et al., 2007), where we train a 3-gram KenLM (Heafield et al., 2013) and learn word alignment by GIZA++ (Och and Ney, 2003). Model weights are tuned on a development set by MERT (Och, 2003).

NMT has better in-domain performance and can generate more fluent texts. We implement the global attentional model proposed by Luong et al. (2015). Detailed hyper-parameters can be found in Section 6.4.2. Note that we do not use Transformer because it empirically works worse

(Zhang et al., 2020b). And we find that the multilingual techniques we explored only significantly improve in-domain performance when using multilingual Bible texts, so we suspect that it biases to Bible-style texts. Hence, we also do not apply multilingual techniques and just train the backbone models with our Cherokee-English parallel texts. We use a 3-model ensemble as our final working model.

Quality Estimation

Supervised QE. The QE (Specia et al., 2010) task in WMT campaign provides thousands of model-translated texts plus corresponding human ratings, which allow participants to train supervised QE models. Fomicheva et al. (2020a) show that supervised models work significantly better than unsupervised ones. Since we are unable to collect thousands of human ratings, we use BLEU (Papineni et al., 2002) as the quality rating. We use 17-fold cross-validation to obtain training data, i.e., we split our 17K parallel texts into 17 folds, use 16 folds to train a translation model, get the translation features plus BLEU scores of examples in the left one fold, repeat this for 17 times, and finally, we get the features plus BLEU scores of 17K examples. Then, we separate 16K examples as a training set to train a BLEU score regressor and evaluate the performance on the left 1K examples. Fomicheva et al. (2020a,b) define three sets of features. However, we need to compute features online, so some features (e.g., dropout features) that require multiple forward computations will greatly increase latency. We use features that will not cause too much speed lag.

For SMT, we use:

1. output length L_t , i.e., the number of words in the translated text;
2. total score;
3. scores of distortion, language model, lexical reordering, phrases penalty, translation model, and word penalty;
4. length normalized (b) and (c) features (i.e., divide each feature from (b) and (c) by (a)).

For NMT, we use:

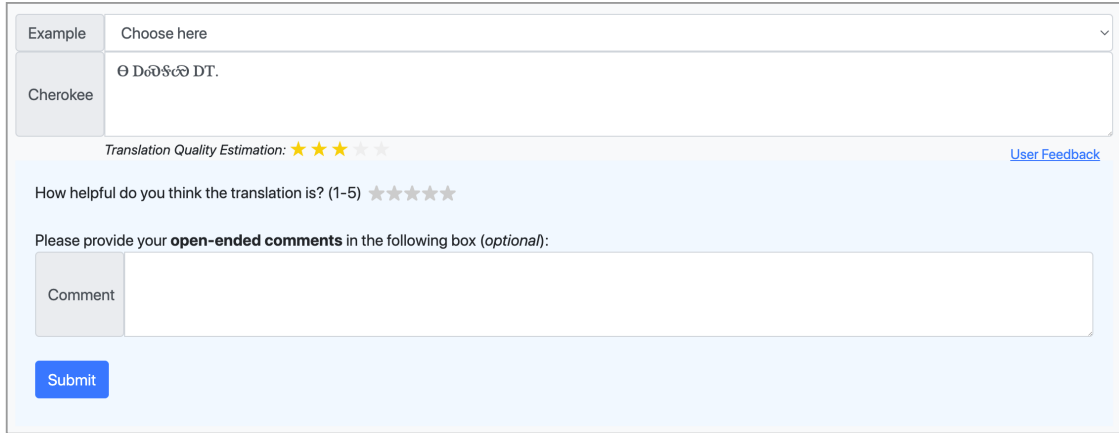
Figure 6.6: Translation interface of our demonstration system. Note that “Ꭰ ᎤᎩᎠᎩᎠᎩ DT.” is not a correct translation. See Figure 6.7 for the corrected translation by an expert.

1. output length;
2. log probability and length normalized log probability;
3. probability and length normalized probability;
4. attention entropy (Fomicheva et al., 2020a,b): $-\frac{1}{L_t} \sum_{i=1}^{L_t} \sum_{j=1}^{L_s} \alpha_{ij} \log \alpha_{ij}$, where L_s is the length of source text, and α_{ij} is the attention weight between target token i and source token j .

Finally, we use XGBoost (Chen and Guestrin, 2016) as the BLEU regressor.¹⁶ As shown in Figure 6.6, we use 5 stars to show QE, therefore, we rescale the estimated quality to 0-5 by dividing the predicted BLEU score (0-100) by 20.

Unsupervised QE. Even though supervised QE works better (Fomicheva et al., 2020a), we suspect that the advantage cannot generalize to open domain scenarios unless we have a large amount of human-rated data to learn from. Hence, we also explore unsupervised QE methods. Unsupervised QE is closely related to uncertainty estimation. We can use how uncertain the model is to quantify how low-quality the model output is. Though it is intuitive to use the output probability as the model’s confidence, Guo et al. (2017) point out that the output probability is often poorly calibrated, so that they propose to re-calibrate the probability on the development set. However, this method is designed for classification tasks and is not applicable for language gen-

¹⁶We also tested GradientBoost (Friedman, 2002) and MLP, but XGBoost empirically works better.



Example Choose here

Cherokee ᎠᎩᎠᎩᎠᎩ DT.

Translation Quality Estimation: ★★☆☆☆ [User Feedback](#)

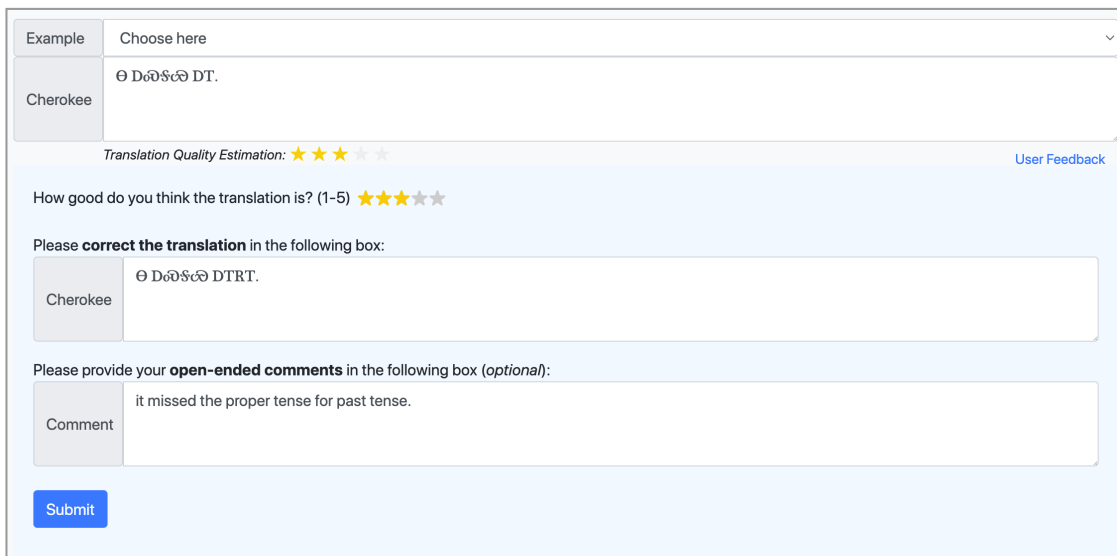
How helpful do you think the translation is? (1-5) ★★☆☆☆

Please provide your **open-ended comments** in the following box (optional):

Comment

Submit

(a) Common User Feedback



Example Choose here

Cherokee ᎠᎩᎠᎩᎠᎩ DT.

Translation Quality Estimation: ★★☆☆☆ [User Feedback](#)

How good do you think the translation is? (1-5) ★★☆☆☆

Please **correct the translation** in the following box:

Cherokee ᎠᎩᎠᎩᎠᎩ DTRT.

Please provide your **open-ended comments** in the following box (optional):

Comment it missed the proper tense for past tense.

Submit

(b) Expert Feedback

Figure 6.7: Two user feedback interfaces of our demonstration system. (b) shows the feedback given by an expert.

eration. Gal and Ghahramani (2016) show that “dropout” can be a good uncertainty estimator, inspired by which Fomicheva et al. (2020b) propose the dropout features. However, the multiple forward passes are not preferable for an online system. Lakshminarayanan et al. (2017) demonstrate that the ensemble model’s output probability can better estimate the model’s uncertainty than dropout. We find that this method is simple yet effective for NMT. Note that we normalize the output probability by the sentence length. Similarly, we rescale the normalized probability (0-1) to 0-5 by multiplying it by 5.

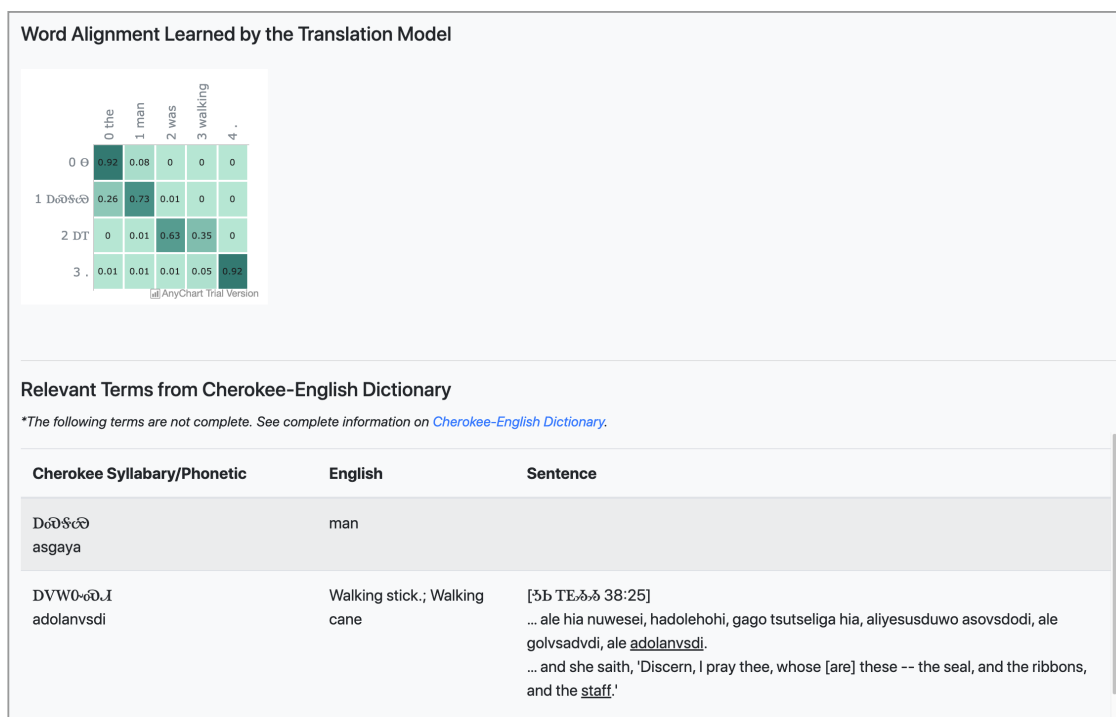


Figure 6.8: Word alignment visualization and link to Cherokee-English Dictionary.

Human Quality Rating. So far, our QE development and evaluation are all based on BLEU. To better evaluate QE performance, we collect 200 human ratings (all rated by Prof. Benjamin Frey, 50 ratings for Chr-En SMT, En-Chr SMT, Chr-En NMT, and En-Chr NMT, respectively. We follow the direct assessment setup used by FLoRes (Guzmán et al., 2019),¹⁷ and thus each translated sentence receives a 0-100 quality rating.

User Feedback & Example Inputs

Enlarging the parallel texts is a fundamental approach to improve the translation model’s performance. Besides compiling existing translated texts, it is important to newly translate English texts to Cherokee by translators. Our system is designed to not only assist these translators but also document their feedback and post-edited correct translation, so that model can be im-

¹⁷0–10: represents a translation that is completely incorrect and inaccurate; 11–29 represents a translation with a few correct keywords, but the overall meaning is different from the source; 30–50 represents a translation that contains translated fragments of the source string, with major mistakes; 51–69 represents a translation that is understandable and conveys the overall meaning of source string but contains typos or grammatical errors; 70–90 represents a translation that closely preserves the semantics of the source sentence; 90–100 range represents a perfect translation.

proved by using this feedback, i.e., human-in-the-loop learning. To achieve this goal, we design two kinds of user feedback interfaces. One is for common users, in which users can rate how helpful the translation is (in 5-point Likert scale) and leave open-ended comments, as shown in Figure 6.7 (a). The other is for experts, in which authorized users can rate the quality, correct the translated text, and leave open-ended comments, as shown in Figure 6.7 (b). Upon submission, we collect 216 pieces of feedback from 4 experts and detailed analysis can be found in Section 6.4.2. Meanwhile, as shown in Figure 6.6, besides inputting text, users can also choose an example input to translate. These examples are from our Cherokee or English monolingual databases. On the one hand, this provides users with more convenience; on the other hand, whenever experts submit translation corrections of an example, we will update its status as “labeled”. Hence, we can gradually collect human translations for the monolingual data.

Other Features

As shown in Figure 6.8, to make model prediction more interpretable to users, we **visualize the word alignment** learned by the translation model. For SMT, we visualize the hard word-to-word alignment; for NMT, we visualize the soft attention map between source and target tokens. Additionally, to provide users with some oracle and handy references from the dictionary, we **link to cherokeedictionary**. We use each of the source and target tokens as a query and list up to 15 relevant terms on our web page.

6.4.2 Evaluation

Implementation Details

Data. To train translation models, we use the 14K parallel data collected by our previous work (Zhang et al., 2020b) plus 3K newly compiled parallel texts. We randomly sample 1K as our development set and treat the rest as the training set. The data is open-sourced at [ChrEn/data/demo](#). To collect human quality ratings, we randomly sample 50 examples from the development set,

			BLEU		Human Rating	
Model		QE	Chr-En	En-Chr	Chr-En	En-Chr
SMT	Supervised	XGBoost	0.75	0.71	0.63	0.44
	Unsupervised	TranslationModel / length	0.36	0.46	0.07	-0.09
		LM / length	0.34	0.43	-0.11	0.11
		PhrasePenalty / length	-0.33	-0.52	0.06	0.03
NMT (ensemble)	Supervised	XGBoost	0.79	0.68	0.53	0.38
	Unsupervised	Exp(LogProbability / length)	0.75	0.63	0.59	0.44
		LogProbability / length	0.45	0.50	0.37	0.52

Table 6.15: Pearson correlation coefficients between QE and BLEU or between QE and human rating. “/ length” represents the normalization by output sentence length.

Model	Chr-En	En-Chr
SMT	17.0	12.9
NMT (single)	18.1	13.8
NMT (ensemble)	19.9	14.8

Table 6.16: The performance of translation models.

and for each of them, we collect 4 ratings for Chr-En/En-Chr SMT and Chr-En/En-Chr NMT, respectively.

Setup. We implement SMT models via Moses (Koehn et al., 2007). After training and tuning, we run it as a server process.¹⁸ We develop our NMT models via OpenNMT (Klein et al., 2017). For both Chr-En and En-Chr NMT models, we use 2-layer LSTM encoder and decoder, general attention (Luong et al., 2015), hidden size=1024, label smoothing (Szegedy et al., 2016) equals to 0.2, dynamic batching with 1000 tokens. Differently, the Chr-En NMT model uses dropout=0.3, BPE tokenizer (Sennrich et al., 2016c), and minimum word frequency=10; the En-Chr NMT model uses dropout=0.5, Moses tokenizer, and minimum word frequency=0. We train each NMT model with three random seeds (7, 77, 777) and use the 3-model ensemble as the final translation model, and we use beam search (beam size=5) to generate translations. We implement the super-

¹⁸<http://www.statmt.org/moses/?n=Advanced.Moses>

vised QE model with XGBoost.¹⁹ XGBoost has three important hyperparameters: max depth, eta, the number of rounds. Tuned on the development set, we set them as (5, 0.1, 100) for Chr-En SMT, (3, 0.1, 80) for En-Chr SMT, (4, 0.5, 40) for Chr-En NMT, and (5, 0.1, 40) for En-Chr NMT. Lastly, the backend of our demonstration website is based on the Flask framework.

Metrics. We evaluate translation systems by BLEU (Papineni et al., 2002) calculated via SacreBLEU²⁰ (Post, 2018). Supervised QE models are developed by minimizing the mean square error of predicting BLEU, but all QE models are evaluated by the correlation with BLEU on development set and the correlation with human ratings. We use Pearson correlation (Benesty et al., 2009).

Quantitative Results

Translation. Table 6.16 shows the translation performance on our 1K development set, which is significantly better than the single-model in-domain translation performance reported in our previous work (Zhang et al., 2020b) and thus achieves the state-of-the-art results. In addition, the 3-model NMT ensemble further boosts the performance.

QE. Table 6.15 illustrates the performance of quality estimation models. In our experiments, we take every feature used in supervised QE as an unsupervised quality estimator. Here, we only present those having a high correlation with BLEU and human rating. It can be observed that, for SMT, supervised QE consistently works better, whereas, for NMT, unsupervised QE has a better correlation with human rating. The obtained correlations with human judgement are moderate ($\gamma \geq 0.3$) to strong ($\gamma \geq 0.5$) (Cohen, 1988). Therefore, we use the trained XGBoost for SMT model’s QE and use the length normalized probability (i.e., $\text{Exp}(\text{LogProbability} / \text{length})$) for NMT model’s QE in our online demonstration system.

¹⁹<https://xgboost.readthedocs.io/en/latest/python/index.html>

²⁰BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0

Qualitative Results

Expert Feedback. Upon submission, we received 216 pieces of feedback from 4 experts (including Prof. Benjamin Frey and 3 other fluent Cherokee speakers). The results are shown in Table 6.17. It can be observed that we received a lot more feedback to NMT than SMT because SMT excessively copies words from source sentences when translating open-domain texts whereas NMT can mostly translate into the target language. On average, there are only 2.3 tokens in the input or translated Cherokee sentence; however, the average translation quality rating is only 2.45 out of 5, which is close to the average rating (43.8 out of 100) of the 200 human ratings we collected. Therefore, according to FLoRes’s rating standard (Guzmán et al., 2019) (see footnote 2), our translation systems *can translate fragments of the source string but make major mistakes* in general. Besides ratings, we received 36 open-ended comments that shine a light on common mistakes made by the models. The most frequent comments are (1) *model gets some parts correct but others wrong*. For example, “it got the subject but not the verb”, “it got the stem right but used 3rd person prefix”, “it missed the part about going to town, but got ‘today’ correct”, etc. (2) *model uses archaic English terms*, like “thy”, “thou”, “speaketh”, etc. because the majority of our training set is the Cherokee Old Testament and the Cherokee New Testament.

Human-in-the-Loop Learning. To improve models based on expert feedback, we propose to simply add the 216 expert-corrected parallel texts back to our training set and retrain the translation models.²¹ The new BLEU results on our development set are 17.3, 13.0, 20.0, 14.8 for Chr-En SMT, En-Chr SMT, Chr-En NMT (ensemble), and En-Chr NMT (ensemble), respectively, which are equal or slightly better than the results in Table 6.16. To tackle the archaic English issue, we simply replace archaic English terms (“thy”, “thou”) with new English terms (“your”, “you”).

²¹We also tried to up-weight these examples by repeating them by 5 or 10 times but did not see better performance.

Model	Chr-En	En-Chr
SMT	12 / 1.92 / 0.39	6 / 2.0 / 0.66
NMT	166 / 2.58 / 0.43	32 / 2.13 / 0.21

Table 6.17: Expert feedback. In each cell, the 3 numbers are the number of feedback received / average quality rating / Pearson correlation coefficient between quality rating and quality estimation.

6.5 Using NLP to Assist in Language Revitalization

6.5.1 Before Diving into NLP Research

We suggest NLP practitioners, who are often “outsiders” of the indigenous communities, three general principles to follow before conducting NLP research on endangered indigenous languages.

Understand and Respect First. Meaningful advances in building speech and language technologies for under-resourced languages hinge upon being able to understand those languages’ speaker communities and their needs. Although the initial temptation among NLP researchers might be to dive in with questions about particular computational tools, that conversation cannot unfold until the speaker communities’ more basic needs are met: *the need for respect, reciprocity, and understanding*. It may be tempting to say “this is outside the scope of our current research,” yet these kinds of behaviors and assumptions are the very behaviors that led to the disenfranchisement of these groups. When we ignore someone’s common humanity and assume that our need for control over the narrative and the situation is greater than their need to be seen and respected, we participate in the same marginalizing and dehumanizing behaviors that led to the problem we are purporting to address. Therefore, it is instrumental that we address the cultural practices and social norms of endangered language communities before assuming we know how to position ourselves, them, and our research within their communities.

Decolonize Research. Decolonizing research is to place indigenous voices and needs in the center of the research process (Smith, 1999; Datta, 2018; Bird, 2020a). As NLP researchers, we are used to certain methodologies. When it comes to questions about endangered languages, it is

tempting for us to formulate the new problems we encounter as what we are familiar with. However, we should always question ourselves: Is the formulation suitable for the language we conduct research on? Are the methodologies we familiar with the only true ways to solve the problems? Unquestioned focus on typical methodologies can make us treat languages as commodities and start to play a “number game” (e.g., the size of the data) and forget the real problem, language revitalization, we intend to solve in the first place (Dobrin et al., 2007). At every research step, it is critical to weigh the burden we put upon the speakers against the benefit that the research can bring back to their community. If the research outcome conveys no new knowledge, information, or benefit to the community, it is no different from “taking” indigenous knowledge that has occurred over the centuries. That is exactly why the word “research” is sometimes the direst (i.e., conjuring up bad memories) word in indigenous world’s vocabulary (Smith, 1999). Finally, it is important to carefully deal with copyright and data governance; meanwhile, we advocate open-sourced and community-contributed works.

Build a Community. Fundamentally, we want to work together with people from the indigenous communities (Bird, 2020a, 2021). It is the most effective way to foster mutual understanding. We should communicate with the indigenous people and get to know their priorities. Common attitudes need to be fostered, common interests need to be found, and common goals need to be set up, before performing the research. These all lead to a community. We would imagine that there is an online community (a website) where native speakers can share their knowledge and language learners can find resources and learn the language together. People can share resources and participate in machine-in-the-loop resource collection projects. NLP researchers can evaluate and share their models in this community. Entertaining language learning or resource collection games can be launched. We hope the community can support wide and sustainable collaborations between indigenous speakers, language learners, and NLP practitioners. Compared to local communities of the speakers, this community will be greatly supported by technologies. A few NLP communities, e.g., MasakhaneNLP (focusing on African languages) and SIGEL (special interest group endangered languages), have been built. Differently, the community we promote here

will support both NLP research and language learning. Lastly, compared to Telegram groups (we are in a few different Telegram groups with Cherokee community members), we want to build a more open community that everyone can have access to.

6.5.2 NLP-Assisted Language Education

Since little inter-generation language transmission is happening, language education is an essential requirement of language revitalization. Computer-assisted language learning has a long-standing history (Higgins, 1983) and two workshops, BEA²² and NLP4CALL²³, are held for research on applying NLP for language education. Here, we discuss three ways in which NLP can potentially assist language education of endangered languages.

Automated Quiz Generation. The most direct way, in which NLP can help, is automatically generating quizzes for language learners. Practicing and producing the language in questions are critical to language acquisition (Gass and Mackey, 2013). Usually, language instructors manually design the quizzes, which is tedious and time-consuming; not to mention, there are not a lot of instructors for endangered languages. However, given the available text of endangered languages, NLP can easily and automatically generate cloze questions. It can also help find distracting wrong answers that happen in a similar context and thus form multi-choice questions (Hill and Simha, 2016; Susanti et al., 2018). To increase playfulness, language learning games, e.g., crossword puzzles and flashcards, can also be automatically generated (Rigutini et al., 2012; Xu and Ingason, 2021). Since these applications involve very basic language processing steps, NLP techniques can be reliably and easily applied.

Automated Assessment. Another widely studied topic is NLP-supported automatic assessment. Though a lot of advanced assessments, e.g., grammar error correction (Bryant et al., 2019), essay grading (Chen et al., 2016), are difficult to be applied for endangered languages, we argue

²²<https://aclanthology.org/venues/bea/>

²³<https://aclanthology.org/venues/nlp4call/>

that some easier assessments are feasible. For example, automatic error analysis and template-based feedback can be provided for language learning quizzes. Another challenging but feasible assessment is to assess the readability or difficulty of language learning materials to provide suitable learning plans for learners of different levels. Using statistic and linguistic features, such as word frequency, morphology or syntactic complexity, etc., readability and difficulty can be automatically predicted (Schwarm and Ostendorf, 2005; Vajjala and Meurers, 2012). However, basic NLP tools, like POS tagger, dependency parser, morphology analyzer, need to be developed before these applications can be realized. The development of these tools requires small but highly-curated data (Blasi et al., 2022).

Community-based Language Learning. Free online language learning platforms that integrate automated quiz generation and assessment have been developed, e.g., Oahpa (Uibo et al., 2015). Taking one step further, we believe that a more effective approach of supporting endangered language education is to build an online and collaborative language learning platform, following the *human computation* technique (Von Ahn, 2008). When using technologies to assist in language revitalization, we often face a dilemma. On the one hand, due to the endangerment, there is not a lot of resources available and it is very expensive (in terms of time, effort, and cost) to collect resources from speakers. On the other hand, machines struggle to reach “useable” and “helpful” performances without a decent amount of training data. *Human computation* aims at combining human and computer to solve problems neither of them could solve alone (Von Ahn, 2008; Garcia, 2013). The most famous example is Wikipedia where Internet users contribute their knowledge together, and incredibly high-quality content has been created. Other successful cases are Duolingo and Tatoeba. Both are for language learners to translate web text and rate each other’s translations. Then, the translated text can serve as learning materials and training data for NLP models. However, Tatoeba only has an English interface, and mixes languages on the same site, making it hard to find peer learners of under-resourced languages. Though Duolingo has language-specific sites, it supports 23 languages so far. Therefore, how to make use of collaborative language learning platforms for endangered languages is a big challenge. Nonetheless, we

believe that it is a promising path to take for teaching endangered languages to the young generation in this information age.

6.5.3 NLP Tools for Cherokee Language Processing

Based on our conversation with a few Cherokee speakers, they agree that some NLP tools are good to have and hold the potentials to be useful in Cherokee language revitalization. Thus, some initial attempts have been made by the Cherokee Language Github group and us (Zhang et al., 2020b, 2021b). Hence, we dive deep into several specific NLP tools for Cherokee language processing in this section. And for any NLP tool we develop, we want to evaluate it by the Cherokee speakers, and we suggest open-sourcing it for free usage. Connecting to our “build a community” proposal, we hope that NLP models can also be shared and used widely and sustainably in the community.

Machine Translation

Ideally, a good machine translation (MT) system can automatically translate the big amount of English text to Cherokee; or it can assist human translators. Dr. David Montgomery, a citizen of Cherokee Nation and a Cherokee language learner, commented on MT: “It would be a great service to Cherokee language learners to have a translation tool as well as an ability to draft a translation of documents for first-language Cherokee speakers to edit as part of their translation tasks. If these tools can be made to work accurately, they would be transformative for the Cherokee language.” Previously, we collected parallel text and developed an MT online translation demo between Cherokee and English (Zhang et al., 2020b, 2021b). However, our system can *translate fragments of the source sentence but make major mistakes*, which is far from being practically useful. The first challenge of MT development is the lack of data. Automatic data mining can help enrich MT training data. But we still need high-quality and diverse evaluation data because existing evaluation sets (Zhang et al., 2020b) are from limited domains (the majority is the Bible). Recently, Flores101, an MT evaluation benchmark covering 101 languages, has been cre-

OCR tools	Original		Screenshot	
	WER	CER	WER	CER
Tesseract	0.355	0.230	0.151	0.063
Google Vision	0.533	0.199	0.468	0.074

Table 6.18: OCR performance of two OCR tools on our evaluation sets. WER: word error rate, CER: character error rates. For both WER and CER, lower is better.

ated (Goyal et al., 2021). Though it has not yet covered Cherokee, we hope it can happen in the future.

The second challenge is processing and producing Cherokee text. Cherokee has rich morphology (see Section 6.2.2). One Cherokee word can be translated into one English sentence. Intuitively, we would think subword tokenization (Sennrich et al., 2016c; Kudo, 2018) is helpful. However, previously, we (Zhang et al., 2020b) showed that applying subword tokenization for English to Cherokee translation is harmful. We argue that it is because we processed Cherokee text in its syllabary rather than in transliterated Latin script, however, morphemes are easier to be learned from the latter. E.g., in ᎠᎵᏍᎦᎵᏍᎦ, *tsaquadvsidev* (when I was growing up), the prefix *ts-* marks relative clauses, but *G* is *tsa*. We suspect that character-level generation (in Latin script) would work better for Cherokee. Additionally, Cherokee has flexible word order that is often determined by whether the information is new or old in relation to the larger discourse (Section 6.2.2). Thus, document-level translations are more reasonable than typical sentence-level translations.

Optical Character Recognition

The majority of Cherokee text is in the format of manuscripts or books, so as many other endangered languages (Joshi et al., 2020b; Bustamante et al., 2020). Though humans can read them, they are not machine-readable, which restricts the flexibility of their use, e.g., automatically creating language learning quizzes. Optical character recognition (OCR) (Smith, 2007) can help extract plain text from PDFs or images. Fortunately, existing OCR tools, like Tesseract-

	audio to phonetic text	audio to syllabic text
WER	0.64	0.21

Table 6.19: The ASR results of finetuned XLSR-53 (Conneau et al., 2020) models. WER: word error rate.

OCR²⁴ and Google Vision OCR API²⁵, support Cherokee and have decent accuracy. However, OCR accuracy is highly influenced by image quality. If the image has a noisy background or the text is surrounded by colorful pictures (which often happens in children books), the OCR accuracy will drop significantly.

To prove this, we create two evaluation sets from Cherokee books (including Cherokee New Testament, children books, Cherokee narratives): (1) *Original* has 20 images, and each image is one complete page from a book; (2) *Screenshot* is obtained by manually conducting screenshots and cutting out text from the 20 images, i.e., removing background noises. For each image in two sets, we manually annotate the corresponding text. Table 6.18 shows the results of Tesseract-OCR and Google Vision OCR API. Both OCR tools achieve significantly lower error rates on the *Screenshot* set than on the *Original* set, which demonstrates the importance of cleaning the images. Tesseract-OCR shows better performance than Google Vision OCR, especially it is better at detecting word boundaries. Although ways to improve image quality are available,²⁶ an easy-to-use tool need to be developed. OCR post-correction methods can also be applied (Rijhwani et al., 2020).

Speech Recognition and Synthesis

Automatic speech recognition (ASR) (Povey et al., 2011) can help language documentation, though indigenous community members may prefer unassisted transcription (Prud'hommeaux et al., 2021). Moreover, ASR holds the potential to automatically transcript audio data and thus

²⁴<https://github.com/tesseract-ocr/>

²⁵<https://cloud.google.com/vision/docs/ocr>

²⁶<https://tinyurl.com/29xnewu9>

	Precision	Recall	F1
Unigram LM	16.6	19.6	17.9
BPE	14.4	16.5	15.4
Morfessor	16.6	16.3	16.5

Table 6.20: The alignment between subwords and gold morphemes.

enrich text corpus. A good amount of Cherokee audio data can be found from the “Cherokee Voices, Cherokee Sounds” radio, Cherokee Phoenix, and recorded meetings. ASR can automatically transcript these audios to produce valuable Cherokee text data. Recently, models that are first pre-trained on audio data and then finetuned on audio-text data have shown great advantages in performing ASR (Baevski et al., 2020). Especially, Conneau et al. (2020) pretrain and finetune a model on 53 languages and release XLSR-53 (supports ASR for 53 languages). It shows reasonable generalizability to unseen and low-resource languages. This sheds light on developing ASR for endangered languages.

Hence, we test its performance for Cherokee ASR. Using the audio-text data open-sourced²⁷ or shared privately by Michael Conrad, we build two ASR models: (1) audio to phonetic text, (2) audio to syllabic text. As shown in Table 6.19, we get surprisingly good performances, especially for the audio-to-syllabic-text model.²⁸ This is very promising, especially when knowing the fact that more self-training strategies can be applied, e.g., pretrain the speech encoder with Cherokee audio data, and more audio-text training data can be compiled. Text-to-speech synthesis (TTS) is more difficult to develop than ASR; nevertheless, following the pretrain-then-finetune paradigm, TTS models for extremely low-resource languages have been introduced (Xu et al., 2020b).

Tokenization and Morphology Parsing

Tokenization is an essential pre-processing step of most NLP models, and it is related to morphology parsing. Subword tokenization has become *de facto* (Sennrich et al., 2016c; Kudo,

²⁷<https://github.com/CherokeeLanguage/cherokee-audio-data>

²⁸The same model finetuned on CommonVoice’s Turkish data gets WER=0.35. <https://tinyurl.com/62eykh9m>

2018). It segments a word into frequent subwords, and subwords are supposed to align with morphemes. Better alignment with morphemes can lead to better downstream performance (Bostrom and Durrett, 2020), while current subword tokenization methods struggle to perform well in morphologically rich languages (Amrhein and Sennrich, 2021).

Here, we evaluate how well subword tokenization can learn real morphemes for Cherokee. We train two subword tokenizers,²⁹ Unigram LM (Kudo, 2018) and BPE (Sennrich et al., 2016c), and one morphology parser, Morfessor (Smit et al., 2014), on our previous MT training set (Zhang et al., 2020b). Instead of using the original syllabic text, we transliterate text into Latin script to make it easier to learn morphemes. We collect gold (expert-labeled) morphemes of 372 Cherokee words from Cherokee Narratives (Feeling, 2018). Then, we use the pretrained tokenizers or parser to tokenize these 372 words and evaluate the alignment between subwords and gold morphemes. As shown in Table 6.20, subwords are poorly aligned with gold morphemes. Nonetheless, Unigram LM (Kudo, 2018) demonstrates better ability of inducing morphemes, which is consistent with the observation made by Bostrom and Durrett (2020). We think better representation methods need to be introduced for Cherokee, and the labeled data from Feeling (2018) can provide supervision.

POS-Tagging and Dependency Parsing

More basic NLP tools like POS tagger and dependency parser are under-developed for Cherokee. These tools can not only support the development of other NLP tools but also be used to predict the readability of language learning materials (Section 6.5.2). Moreover, data for these tasks can serve as language learning materials for understanding Cherokee linguistics. Though unsupervised methods have been proposed (Stratos et al., 2016; Kim et al., 2019), usually small but high-quality labeled data, like Universal Dependencies (Nivre et al., 2016), is needed (Blasi et al., 2022). Therefore, data annotation by experts is required and community-based data collection strategies can be applied. Moreover, the parallel English data and English tagger/parser can

²⁹We use SentencePiece (Kudo and Richardson, 2018).

assist the annotation on the Cherokee side, which will also produce English-Cherokee word/phrase-level alignments as by-products. These alignments are valuable Cherokee language education resources, e.g., asking students when you have “structure X” in English, what is the corresponding “structure Y” in Cherokee?

6.6 Conclusion

In this line of research work, we first collected a Cherokee-English translation dataset and investigated various translation systems to support the translation between Cherokee and English. Then, we developed the first online Cherokee-English machine translation demo, which supports not only translation but also quality estimation and collecting human feedback. Finally, we “zoomed out” from the translation task, reviewed the big picture of using NLP to assist in language revitalization, and discussed other valuable NLP tools for Cherokee and the challenges of developing them. We hope our work can encourage future work to think and plan the path forward for Cherokee language processing as well as language processing for other underrepresented languages.

CHAPTER 7: LANGUAGE IMBALANCE IN MULTILINGUAL TOKENIZER TRAINING

7.1 Introduction

Tokenization is an essential pre-processing step for most natural language processing (NLP) models. Out of different tokenization methods, subword tokenization (Schuster and Nakajima, 2012; Sennrich et al., 2016c; Kudo, 2018) has become *de facto*. The creation of each subword is mainly based on frequency, i.e., if two characters often appear together, they will be merged into a subword. When more than one language is involved, instead of learning independent tokenizers for each language, people usually train a joint tokenizer from a multilingual training corpus (Sennrich et al., 2016c; Devlin et al., 2019). In this case, the data percentage of each language directly affects how it will be represented. If one language dominates the training corpus, its words will mostly stay intact and hardly be split into subwords. In contrast, if the language gets starved, it will be excessively tokenized into characters, thus, the sentence length will be dramatically longer, and some tokens will be considered as unknown (UNK). Moreover, Neural machine translation (NMT) is known to be bad at dealing with long sentences and UNKs (Koehn and Knowles, 2017).

Recently, there is an increasing interest in building multilingual neural models that can process multiple languages (Devlin et al., 2019; Liu et al., 2020; Xue et al., 2021b). A challenge that comes with this important task is to balance languages with different amounts of training data to avoid low-resource languages being under-represented, e.g., being excessively tokenized and being less seen by the neural models. Existing works usually adopt the *temperature sampling* strategy (Devlin et al., 2019; Arivazhagan et al., 2019; Conneau and Lample, 2019; Xue et al., 2021b) (see detailed descriptions in Section 7.2.2). However, very few investigations of how lan-

guage imbalance affects downstream performance have been conducted. Additionally, whenever previous works apply a certain language balancing strategy, they apply it for both *tokenizer training* (balancing the data sizes of different languages in the tokenizer training corpus) and *model training* (balancing the frequencies of sampling training mini-batches from different languages). Until now, it is unclear how each of them separately affects the downstream performance.

In this work, we specifically investigate how robust NMT is to language imbalance in tokenizer training. We propose to vary the data ratio among languages in the tokenizer training corpus while keeping other settings (e.g., language sampling for model training, hyperparameters) fixed, and then check how translation results change (Section 7.3.1). However, finding the best data ratio through performing the downstream task is highly expensive. To provide an easy indication of tokenizer quality (or early prediction of downstream performance), we examine two intermediate features (Section 7.3.2): *UNK rate* – the average percentage of unknown words (marked with the UNK token) in each sentence, and *closeness to the character level* – the average sentence length in subwords divided by sentence length in characters.

Through comprehensive bilingual and multilingual experiments among 8 languages (English, Tagalog, Icelandic, Danish, Indonesian, Tamil, Greek, and Chinese), we make the following **five main observations**: (1) NMT performance is more robust to language imbalance than we usually expected: especially when languages share scripts, performance drops only happen when the data ratio of two languages is as disparate as $1:10^5$. (2) Better performance is often achieved when languages are more balanced: we observe moderate Pearson correlations between translation performance and the degree of language balance. (3) English can “never” be starved because English tokens often appear in the “monolingual” data of other languages. (4) In most cases, the two features (UNK rate and closeness to the character level) can hint at poor translation performances before performing the task. (5) NMT is more sensitive to language imbalance in model training than in tokenizer training. See more observations and discussions in Section 7.3 and Section 7.4.

Based on these observations, we provide the following **two practical suggestions**: (1) Instead of using temperature sampling, we want to keep the involved languages as balanced as pos-

sible when training a new multilingual tokenizer; (2) Before applying a pretrained tokenizer for new experiments or languages, we suggest evaluating it on a development set to make sure every language’s UNK rate is low (lower than around 3.7%, according to our experiments) and every language’s closeness to the character level is also low (lower than around 0.87, according to our experiments).¹

7.2 Background and Related Work

7.2.1 Tokenization Methods

Over the years, many tokenization methods have been proposed. Early works tokenize texts into “words”, e.g., `MosesTokenizer` (Koehn et al., 2007). However, language-specific tokenizers are needed and it often ends up with many rare tokens or UNKs. *Subword tokenization* methods were introduced to tackle this problem: the idea is to keep frequent words intact and split rare words into frequent subwords. Subword tokenization has become *de facto*. Schuster and Nakajima (2012) introduce `WordPiece` that starts from all characters and gradually merges two units that improve language model (LM) likelihood the most. Sennrich et al. (2016c) propose to learn subwords via Byte-Pair Encoding (BPE) that merges the most frequent pairs first. Kudo (2018) propose a unigram LM method. It starts with a large vocabulary and gradually prunes it down to the desired size by removing tokens that are less likely to reduce the unigram LM likelihood. Subword tokenization methods usually assume the existence of pre-tokenization (e.g., split by whitespaces), which can cause de-tokenization ambiguity. To address this, `SentencePiece` (Kudo and Richardson, 2018) treats whitespace as a special symbol, `_` (U+2581), to achieve *lossless* tokenization. This toolkit supports both BPE and unigram LM tokenization. Despite the success of subword tokenization, it is no panacea, e.g., it is out-of-the-box and agnostic to the downstream tasks, it has no guarantee that subwords are meaningful, and it is vulnerable to typos (Sun

¹The exact threshold numbers (3.7% and 0.87) are based our experiments and may not always hold. But we believe that the concept of checking the two features (UNK rate and the closeness to the character level) to make sure they are low enough should generalize to other situations.

et al., 2020). Thus, “tokenization-free” models that directly encode characters or bytes or visuals have been introduced (Chung et al., 2016; Lee et al., 2017; Salesky et al., 2021) and are gaining more interest recently (Clark et al., 2022; Xue et al., 2021a; Tay et al., 2021).

7.2.2 Multilingual Tokenization

Along with the development of multilingual models, people start to deal with multilingual tokenization. Firat et al. (2016) learn a 30K subword vocabulary for each language. Johnson et al. (2017) oversample languages to the same size and train a joint WordPiece vocabulary. Recent multilingual works adopt this joint-vocabulary method, but instead of oversampling languages to the same size, they use *temperature sampling* which was first introduced by multilingual BERT (mBERT) (Devlin et al., 2019). Given the original data distribution $\{p_i\}_{i=1}^N$, where p_i is the percentage of the i^{th} language out of the total N languages, they exponentiate each p_i by a factor S ($0 \leq S \leq 1$), i.e., p_i^S . Then, they re-normalize them to get the new percentage of each language $\hat{p}_i = p_i^S / \sum_i p_i^S$, and they sample data according to the new percentages. Essentially, it down-samples high-resource languages and up-samples low-resource ones. Arivazhagan et al. (2019) redefine S as $\frac{1}{T}$ (T stands for temperature). S is usually set around 0.2 to 0.7, i.e., *flattening the data distribution to some degree but not to uniform distribution* (Arivazhagan et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020; Xue et al., 2021b). Chung et al. (2020) challenge this joint vocabulary recipe and propose to learn separate vocabularies for each language cluster.

7.2.3 Analysis and Assessment of Tokenization

Since the choice of tokenization algorithm and training parameters affects downstream performances, previous works try to analyze or assess tokenization. Some works focus on the choice of vocabulary size. Gowda and May (2020) show that the near-optimal vocabulary size is when 95% of tokens appear more than 100 times in the training set. Ding et al. (2019) find that low-resource language pairs usually require fewer than 4K BPE merge-operations. Xu et al. (2021) evaluate vocabularies by Marginal Utility of Vocabularization and propose to tokenize as well as

find the optimal vocabulary size via the Optimal Transport method. Some other works compare different tokenization algorithms. Domingo et al. (2018) compare 5 tokenizers and the best tokenizer varies across language pairs. Bostrom and Durrett (2020) compare BPE to unigram LM for LM pretraining and show that unigram LM learns subwords that align better with morphology and leads to better performance.

When multiple languages are involved, Gerz et al. (2018) show that language typology is correlated with LM performance. Ács (2019) find that mBERT (Devlin et al., 2019) vocabulary are dominated by subwords of European languages, and the tokenizer keeps English mostly intact while generating different distributions for morphologically rich languages. Rust et al. (2021) observe that mBERT usually performs worse than its monolingual counterparts because language-specific tokenizers keep the language from being excessively tokenized. Some works compare different *temperature sampling* factors (S or T). Arivazhagan et al. (2019) compare multilingual translation results of using temperature $T=1, 5, 100$, and find that $T=5$ works best. Xue et al. (2021b) compare multilingual LM performances for sampling factor $S=0.2-0.7$ and find that $S = 0.3$ is the best. However, note that the performance difference is a joint effect of both tokenizer and model training because the sampling is applied for both. Differently, in this paper, we analyze how language imbalance specifically in multilingual tokenizer training affects the downstream translation performance.

7.3 Bilingual Experiments

To examine how language imbalance in tokenizer training affects downstream translation performance, we first conduct English-centric bilingual experiments in which imbalance only happens for one single pair of languages (i.e., English and another language). This gives us a more controlled setting compared to when multiple languages are involved. Nonetheless, we conduct multilingual experiments in Section 7.4. Our main methodology is to keep the total tokenizer training data size fixed, gradually “starve” English, i.e., reduce English data percentage and increase the percentage of the other language, and then check the downstream translation perfor-

Language	Code	Script	En-* bitext	Mono. text
English	en	Latin	-	2B
Tagalog	tl	Latin	71K	107M
Icelandic	is	Latin	1M	37M
Danish	da	Latin	11M	343M
Indonesian	id	Latin	39M	1B
Tamil	ta	Tamil	97K	68M
Greek	el	Greek	24M	200M
Chinese	zh	Han	38M	293M

Table 7.1: 8 languages in our experiments. K/M/B stands for thousand/million/billion. Mono. stands for monolingual. Numbers are the number of sentences (pairs).

mance. It is important to note that, to separate the influences of tokenizer and model, we use different data for tokenizer training and model training, and the model training data are always the same.

7.3.1 Experimental Setup

Languages. We experiment with 8 languages: English (en), Tagalog (tl), Icelandic (is), Danish (da), Indonesian (id), Tamil (ta), Greek (el), Chinese (zh). The data statistics are shown in Table 7.1. According to Flores101 (Goyal et al., 2021), Icelandic, Tamil, and Tagalog are *low-resource* ($\leq 1\text{M}$ bitext), while Danish, Greek, Chinese, and Indonesian are *mid-resource* ($\leq 100\text{M}$ bitext). Tagalog, Icelandic, Danish, and Indonesian are Latin languages and thus share scripts with English; while Tamil, Greek, and Chinese are non-Latin.

Translations. We conduct English-centric bilingual translations in 14 directions: en-tl, tl-en, en-is, is-en, en-da, da-en, en-id, id-en, en-ta, ta-en, en-el, el-en, en-zh, zh-en. We train one translation model for each direction.

Variables. For each translation direction, we have the following controlled, independent, and dependent variables:

Controlled variables:

Tokenizer training data: We use the same monolingual data as Flores101 (Goyal et al., 2021). The total monolingual data sizes of each language are listed in Table 7.1. We sample from these monolingual datasets to get the desired tokenizer training data size.² We keep the total tokenizer training data size as 2M, which contains $x\%$ English data and $1 - x\%$ data of the other language.

Tokenizer parameters: We use SentencePiece model (SPM) with unigram LM algorithm (Kudo, 2018; Kudo and Richardson, 2018). We set vocabulary size as 5K,³ total training data size as 2M, and character coverage as 0.99995 (or 0.995 when Chinese is involved because Chinese has a richer character set).

Translation training data: We also use the same parallel data as Flores101 (Goyal et al., 2021) (data sizes are in Table 7.1). As mentioned above, we do not change this model training data across different experiments. And following previous works (Section 7.2.2), we always use temperature sampling with $S = 0.2$ for model training.

Translation evaluation data: We evaluate on Flores101 (Goyal et al., 2021) dev sets and report results on its devtest sets.

Translation model: Transformer (Vaswani et al., 2017) with 12-layer encoder and 12-layer decoder (Transformer 12-12).

Model training and testing hyper-parameters: Adam optimizer (Kingma and Ba, 2015), learning rate = 0.001, and beam size = 5. See more implementation details in Section 7.5.

Independent variable:

*English data percentage in 2M tokenizer training data*⁴: we experiment with 9 different percentages (0%, 0.001%, 0.1%, 10%, 50%, 90%, 99.9%, 99.999%, 100%). E.g., if we conduct en-zh/zh-en translations with English percentage=0.001%, there are 20 English sentences and 2M -

²To minimize sampling influence, we shuffle each monolingual dataset once and then always sample the first X sentences.

³We set vocabulary size as 5K because (1) a small vocab size makes the “competition” between languages more “fierce” and thus makes it easier to show the problem of language imbalance, and (2) it resembles a multilingual setting: Flores101 uses a 256K vocabulary for 101 languages – 2.5K tokens per language on average.

⁴We choose to directly vary the data percentage rather than sampling temperature because it grants us the flexibility of making high-resource languages hypothetically low-resource and experimenting with extreme data ratios (100%: 0%).

20 Chinese sentences in SPM tokenizer training data. Hence, for each translation direction, we have 9 experiments with 9 different vocabularies. See examples of how sentences are tokenized at different English percentages in Table 7.4.

Dependent variable:

Translation performance: we evaluate it by sentence-piece BLEU (spBLEU) (Goyal et al., 2021)⁵ and chrF (Popović, 2015). Metrics are computed by SacreBLEU (Post, 2018).⁶ We report the 3-seed average for each experiment.

7.3.2 Intermediate Features

Previous works have shown that without training downstream models, some intermediate features can be good indicators of the tokenizer’s quality (Gowda and May, 2020; Chung et al., 2020; Xu et al., 2021). In this work, as the English data percentage varies, either English or the other language will get starved – sentence lengths will become longer and unknown words (UNKs) will appear. Hence, we examine the following two features:

Closeness to the character level, defined as the average $\frac{\text{sentence length in subwords}}{\text{sentence length in characters}}$. Some languages may intrinsically have longer sentence lengths than others. To be comparable across languages, we normalize it by the upper bound – sentence length in characters.

UNK rate, which is defined as the average $\frac{\text{number of UNKs}}{\text{sentence length in subwords}}$. Note that when the UNK rate increases, long unknown tokens will not get split into subwords, and thus the sentence length will be shorter and the closeness to the character level will decrease.

The first two columns of Figure 7.1 illustrate how the intermediate features change as the English data percentage changes. The first row (a) shows features of the 4 Latin languages, while the second row (b) is those of the 3 non-Latin languages. Note that both features are computed on Flores101 (Goyal et al., 2021) devtest sets.

⁵Computing BLEU (Papineni et al., 2002) requires a tokenizer. However, not all languages have language-specific tokenizers available. spBLEU (Goyal et al., 2021) unifies the evaluation across languages by first tokenizing languages via a 256K multilingual SPM and then computing BLEU.

⁶https://github.com/ngoyal2707/sacrebleu/tree/adding_spm_tokenized_bleu

Closeness to the character level. In Figure 7.1 (a), as the English percentage increases, the closeness to the character level of English (gray markers) decreases while that of other languages (markers with other colors) increases. It is because when the English percentage gets larger, the other language’s tokens will become rarer and be excessively tokenized into subwords. Differently, in Figure 7.1 (b), though the trend of English stays the same, the trend of other languages first increases close to 1.0 and then decreases because UNKs start to appear. Even when English occupies 100%, Latin languages still have sentence lengths much shorter than the sentence length in characters because they share scripts with English. In contrast, each of the 3 non-Latin languages reaches close to the character level at a certain point. English never have very long sentence lengths.

UNK rate. In Figure 7.1 (a), most UNK rates are trivial (close to 0), except that Icelandic (is) and Danish (da) have non-trivial UNK rates when English percentage $\geq 99.999\%$. In Figure 7.1 (b), all three non-Latin languages have very high UNK rates after the English percentage increases to a certain point. For example, Chinese (zh) has a 45.7% UNK rate at English=99.9%, and it is when its closeness to the character level drops dramatically. English always has trivial UNK rates.

7.3.3 Translation Results

The second two columns of Figure 7.1 shows how the translation results change as the English data percentage changes. The first row (a) shows spBLEU and chrF scores of the 4 Latin languages, while the second row (b) are those of the 3 non-Latin languages. We obtain the following takeaways.

NMT performance is quite robust to language imbalance especially when languages share scripts. It can be observed from Figure 7.1 (a) that the performance stays quite stable across all English percentages for Latin languages. Performance drops only happen for English to Icelandic (en-is) and English to Danish (en-da) at extremely high English percentages ($\geq 99.999\%$), i.e., only 20 Icelandic or Danish sentences are in the 2M tokenizer training data. And it still does not

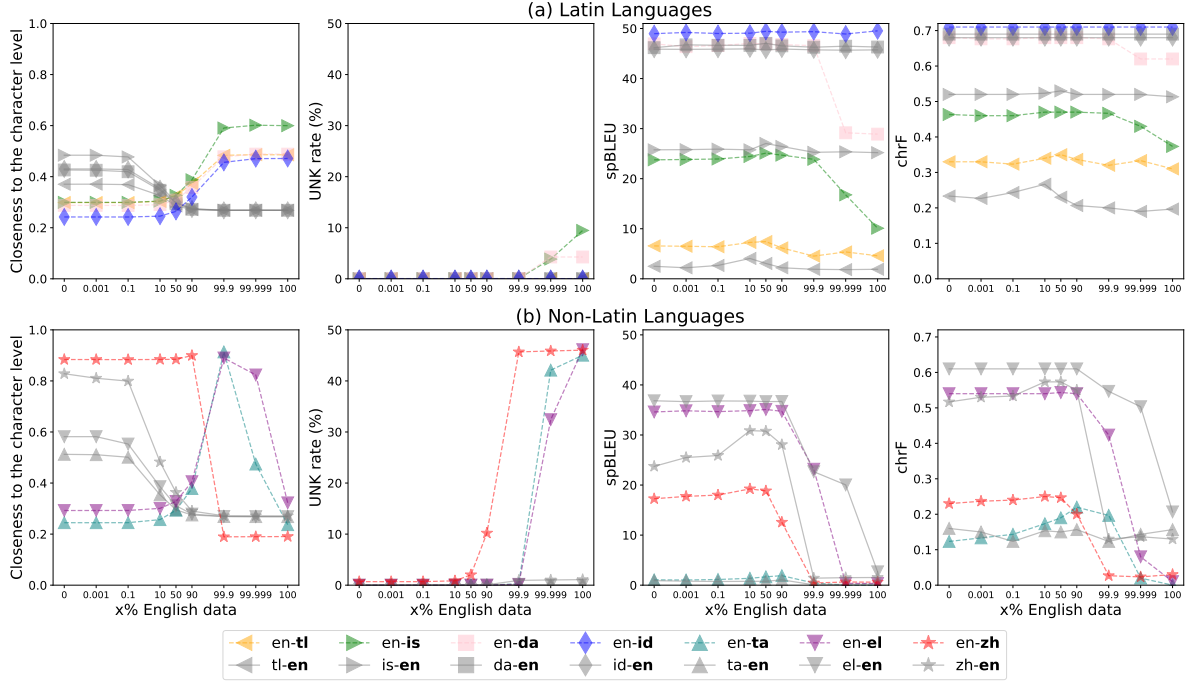


Figure 7.1: Results of our main bilingual experiments. Marker shapes denote the language pairs; dash or solid lines represents out-of-English or into-English directions; colors are for each target language. E.g., $--\triangleleft$ (en-ta) denotes Tamil features (*Closeness to the character level* or *UNK rate*) or English to Tamil translation results (*spBLEU* or *chrF* scores); $-\triangleleft$ (ta-en) represents English features or Tamil to English translation results. X axes are in log10 scale.

affect the translation performances of is-en and da-en. Differently, in Figure 7.1 (b), the performance is less stable for non-Latin languages, but drops still happen when the English percentage is $\geq 90\%$. English to Chinese (en-zh) drops at English=90%. English to Tamil (en-ta)⁷ and English to Greek (en-el) both drop at English=99.9%. Similarly, into-English directions are more stable and get worse later (at higher English percentages). Surprisingly, in both (a) and (b), the translation performance usually stays stable or drops less significantly as the English percentage decreases to 0%.

Better performance is often achieved when languages are more balanced. Out of the 14 translation directions, 12 directions get the best spBLEU scores between English=10% to English=90%. We evaluate the Pearson correlation between spBLEU scores and *data ratios* of two languages. The data ratio is 1 when English=50%, and it is 0 when English=0% or 100%, i.e.,

⁷Note that at English=99.9%, Tamil’s chrF scores only drop slightly while its spBLEU scores drop more significantly (en-ta drops from 1.9 to 0.4 and ta-en drops from 1.1 to 0.1).

the more balanced the two languages are, the higher the data ratio is. The average correlation across 14 directions is 0.38 (moderate correlation (Cohen, 1988)). Thus, we are more likely to get a good performance when languages are more equally sampled.

English can “never” be starved. Initially, we were expecting a symmetric trend, i.e., if the performance drops as the English percentage increases, it should also drop when the percentage decreases. However, as shown in Figure 7.1, for both Latin and non-Latin languages, the performance stays relatively stable as the English percentage decreases to 0%. We suspect that other languages’ monolingual data contains many English words. First, we find that about 3.6% and 2.6% characters in Tamil and Chinese monolingual data are English characters (a-zA-Z) respectively. Then, we remove all English characters from Tamil or Chinese monolingual data and re-conduct the experiments of English=0.001%. English-Tamil/Tamil-English spBLEU scores reduce from 1.0/0.8 to 0.0/0.3. Similarly, English-Chinese/Chinese-English spBLEU scores drop from 17.7/25.5 to 0.2/0.1. Hence, the results support our hypothesis.

Closeness to the character level and UNK rate can warn of poor downstream performance.

We find that the translation performance usually drops greatly when the two features surpass some thresholds. As shown in Figure 7.1 (a), both English to Icelandic (en-is) and English to Danish (en-da) get noticeably worse at English=99.999%, and it is exactly when Icelandic and Danish have non-trivial UNK rates (3.9% for is and 4.3% for da). Similarly, in Figure 7.1 (b), English to Chinese (en-zh) deteriorates at English=90% when Chinese UNK rate is 10.2%. English to Tamil (en-ta) and English to Greek (en-el) both drop at English=99.9% when they have trivial UNK rates but their closeness to the character level are 0.91 and 0.89 respectively. Additionally, we examine whether the same pattern can still be observed when getting the features on a different evaluation set. We get features from the dev set and a subset of our training set (5000 sentence pairs). As Figure 7.2, despite the slightly lower thresholds (3.7% UNK rate and 0.87 closeness to the character level), the same trends are observed. Hence, we suggest checking these two features on an evaluation set before performing the task. Poor translation performances are

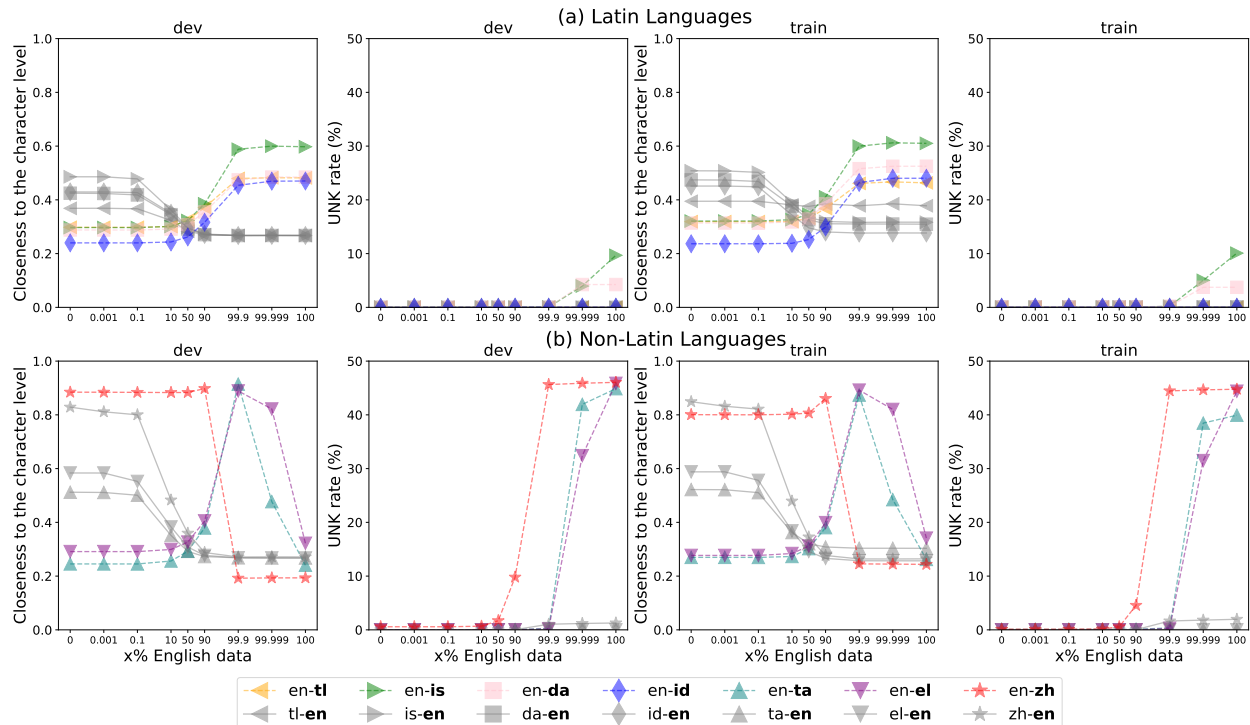


Figure 7.2: In each row, the first two subplots are features computed on the Flores101 dev set; the second two subplots are features computed on a subset of our training set. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.

likely to be obtained when any language’s UNK rate is larger than around 3.7% or its closeness to the character level is larger than around 0.87.

7.3.4 Ablations

Here, we want to verify our takeaways under several different experimental settings.

Reducing the translation model size or using BPE does not affect the robustness to language imbalance. Model capacity can affect its robustness. Hence, we replace our default Transformer 12-12 (Vaswani et al., 2017) model with a smaller model, Transformer 6-6 (6-layer encoder and 6-layer decoder). The intermediate features are the same as Figure 7.1, and the translation results are illustrated in Figure 7.3. It has exactly the same trends as for the larger model (Figure 7.1). In addition, we verify if our takeaways can generalize to a different tokenization algorithm, BPE (Sennrich et al., 2016c). Figure 7.4 shows that BPE gets very similar performances to unigram

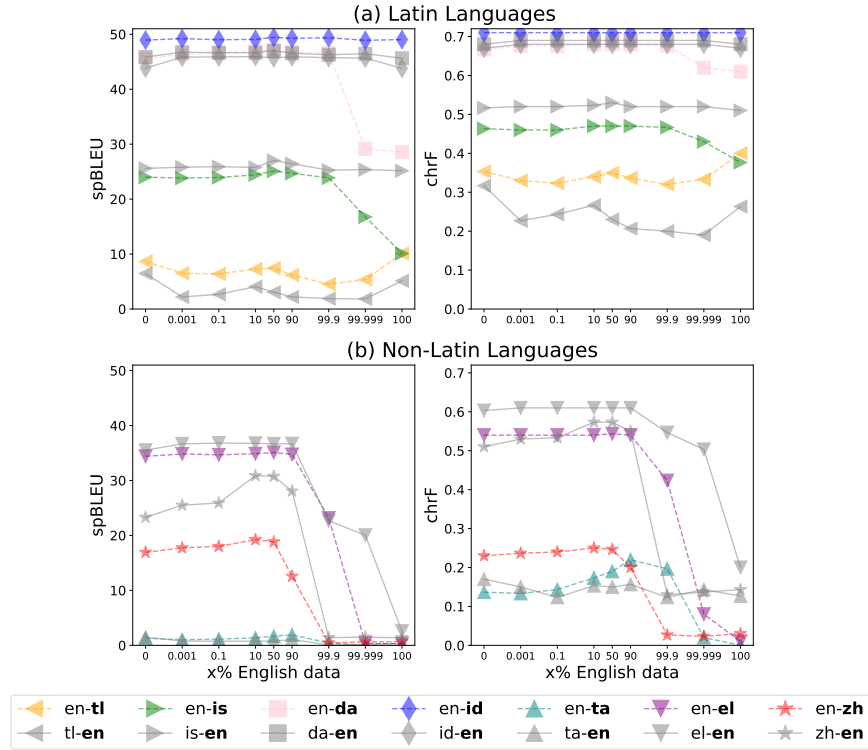


Figure 7.3: Translation results of bilingual experiments with a smaller model (Transformer 6-6). Markers share the same meanings as Figure 7.1. X axes are in log10 scale.

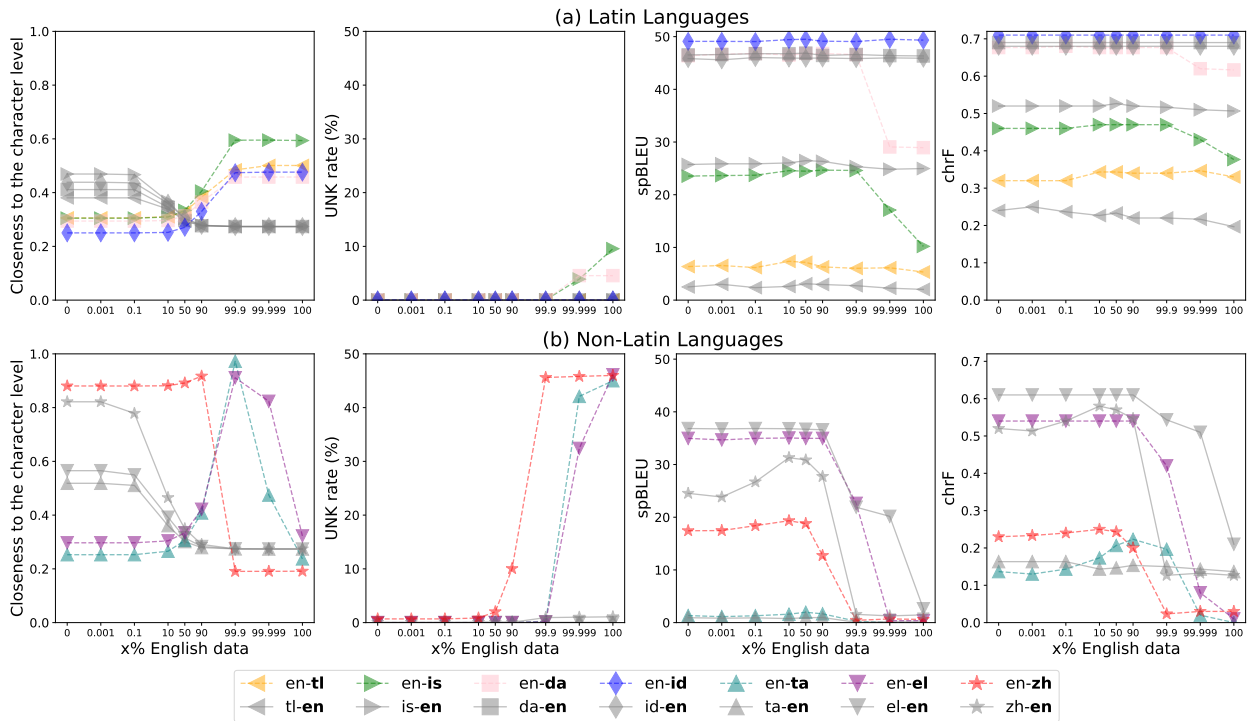


Figure 7.4: Intermediate features and translation results of bilingual experiments with a BPE tokenizer. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.

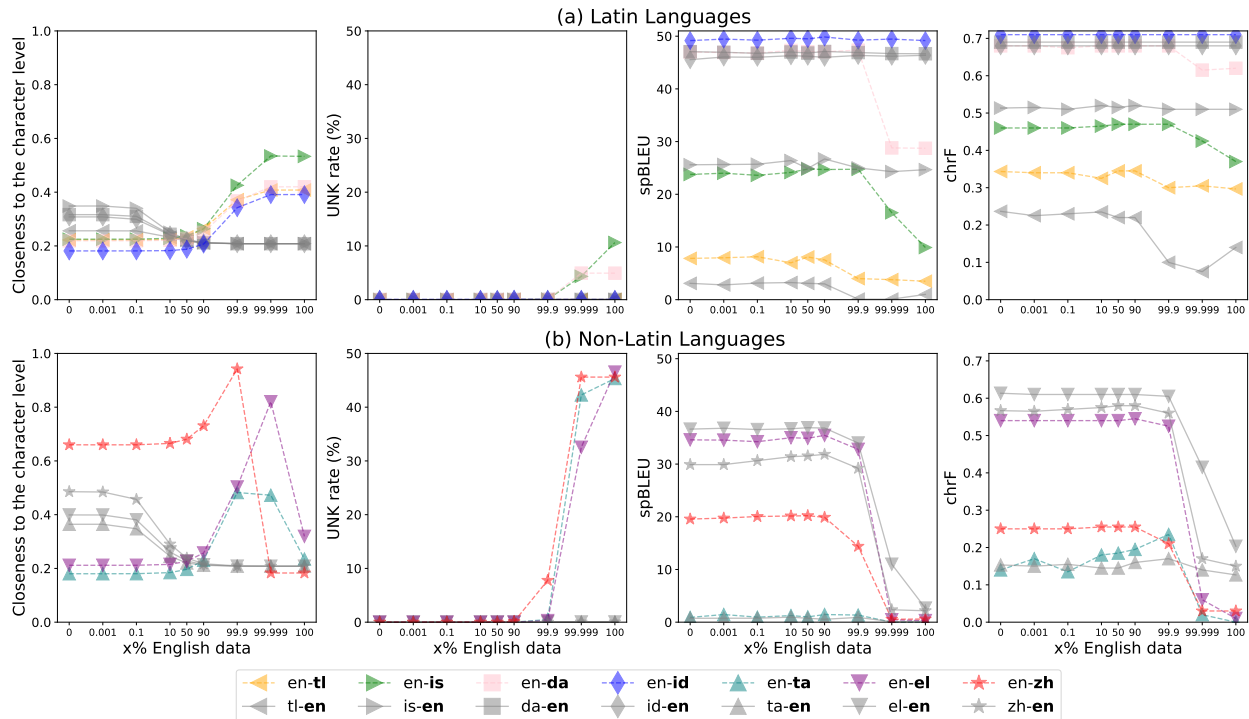


Figure 7.5: Intermediate features and translation results of bilingual experiments with a 32K vocabulary. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.

LM across all translation pairs. The same trends are also observed as Figure 7.1 but with slightly higher thresholds.

Increasing the vocabulary size can improve the robustness when languages do not share scripts. Our default vocabulary size is 5K because it simulates a multilingual setting (see footnote2). However, earlier works used a larger vocabulary for bilingual experiments (Firat et al., 2016). Intuitively, a larger vocabulary can be more robust to language imbalance because it has a larger capacity to include more infrequent words. Hence, we test a 32K vocabulary, and results are shown in Figure 7.5. Compared to Figure 7.1, it has two distinctions: (1) For non-Latin languages, performance drops happen later: English to Chinese drops at 99.9% (instead of 90%) when Chinese UNK rate is 7.8%; English to Tamil and English to Greek both deteriorate greatly at 99.99% (instead of 99.9%) when Tamil and Greek UNK rates are 42.3% and 32.5% respectively; (2) Surprisingly, translations between English and Tagalog perform obviously worse when English \geq 99.999%, despite Tagalog’s trivial UNK rate and short sentence length. Overall, increas-

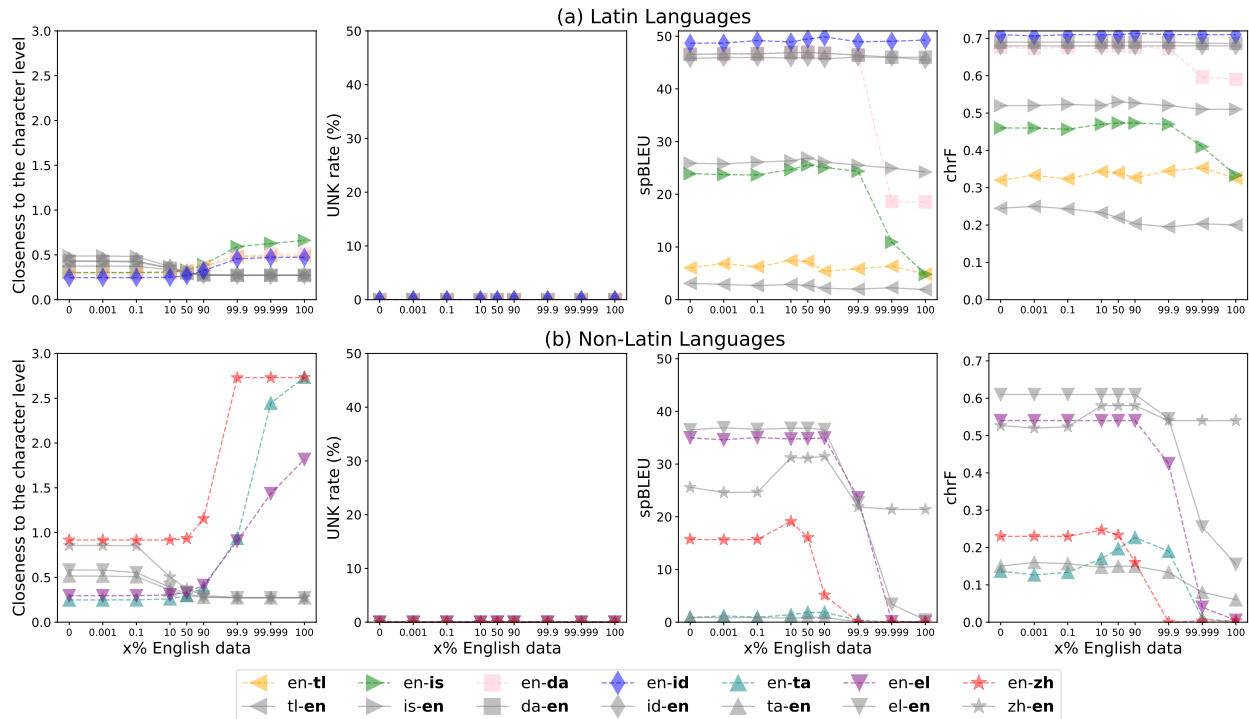


Figure 7.6: Intermediate features and translation results of bilingual experiments with byte-fallback. Note that here the UNK rates are all 0, and closeness to the character level can be larger than 1 because one character can be represented by multiple bytes. Markers share the same meanings as Figure 7.1. X axes are in log10 scale.

ing the vocabulary size improves the robustness to language imbalance for translations between English and non-Latin languages but not for that between English and Latin languages.

Applying byte-fallback does not improve the robustness. Here, we apply the “byte-fallback” feature of SentencePiece (Kudo and Richardson, 2018) which uses 256 UTF-8 bytes to represent unknown characters and thus eliminates UNKs. Figure 7.6 illustrates the results. As expected, UNK rates are all 0, while closeness to the character level can be larger than 1 because one character can be represented by multiple bytes. For Latin languages, noticeable drops still only happen for Icelandic and Danish starting from 99.999%, but differently, they have 0 UNK rates and not high closeness to the character level (0.65 and 0.53). Moreover, performance drops are surprisingly more dramatic compared to Figure 7.1. The performances of all 3 non-Latin languages get worse at the same percentages as Figure 7.1, and the drop is more significant for Greek to English while less significant for Chinese to English. Overall, applying byte-fallback does not improve the robustness reliably.

	100%	100%+char	best
en-ta	0.0	0.1	1.9
ta-en	0.2	0.1	1.1
en-el	0.3	18.6	35.1
el-en	2.7	18.5	36.7
en-zh	0.6	20.0	19.2
zh-en	1.5	31.2	30.9

Table 7.2: Translation results (spBLEU scores) of adding the non-Latin language’s characters to the vocabulary at English=100% (**100%+char**). For comparison, the **100%** column shows the results before adding characters and the **best** column shows the best results out of all percentages.

When English=100%, adding characters of the non-Latin language to the vocabulary can

improve the performance. When English occupies 100% of the tokenizer’s training data, the tokenizer only “knows” English. Other Latin languages share scripts with English, so it shows surprisingly good generalizability. However, for non-Latin languages, near all tokens are UNKS, and thus translation performances are very poor. We wonder how much the performance will increase by simply adding the characters of the non-Latin language to the vocabulary. We conduct this experiment for each of the 3 non-Latin languages, and the results are shown in Table 7.2.

Compared to the original setting (100%), adding characters (100%+char) dramatically improves the performance except for ta-en. Despite that, for Tamil or Greek, it works greatly worse than the best we can achieve when Tamil or Greek data involves in tokenizer training. But, for Chinese, it outperforms the best results probably because one Chinese character is usually one “word”.

Examples. Table 7.4 are examples of how sentences in English, Indonesian, and Chinese are tokenized at different English percentages.

7.4 Multilingual Experiments

Here, we move to a more complex multilingual setting. Similarly, we want to understand how the data percentages of the involved languages affect their downstream translation performance.

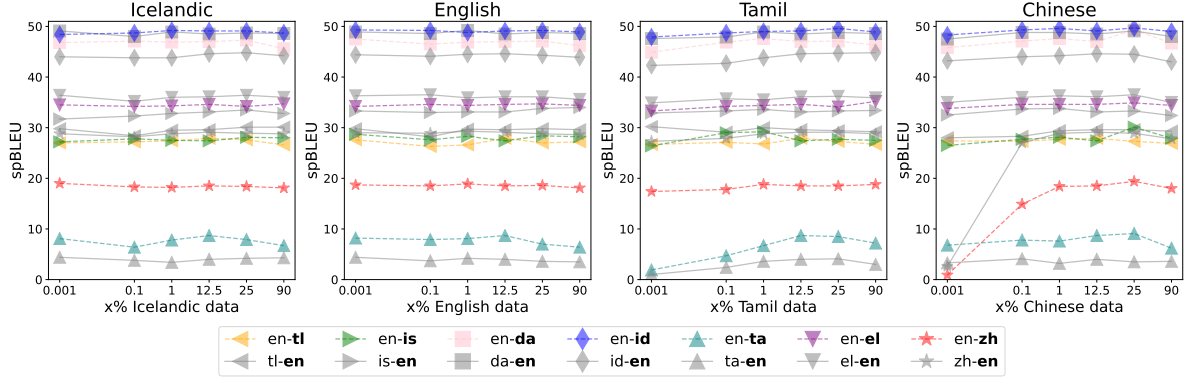


Figure 7.7: Translation results (spBLEU) of our main multilingual experiments. Marker shapes denote the language pairs (though all pairs share the same NMT model); dash or solid lines represents out-of-English or into-English directions; colors are for each language. E.g., $--\blacktriangle--$ (en-ta) denotes English to Tamil translation results; $-\blacktriangle-$ (ta-en) represents Tamil to English translation results. X axes are in log10 scale.

7.4.1 Experiment Setup & Features

We still experiment with the 8 languages and the 14 translation directions, as introduced in Section 7.3.1. Differently, we use one model (Transformer 12-12) to conduct all the 14 translations at the same time. As a result, the model capacity for each translation direction is dramatically reduced. Most of the *controlled variables* stay the same as Section 7.3.1, except that we increase the vocabulary size to 20K (maintaining around 2.5K per language) and increase the total tokenizer training data size to 10M. Since here we have 8-language data to train the tokenizer, we can not use the old *independent variable*. Instead, we propose to first choose one language and then vary its percentage (0.001%, 0.1%, 1%, 12.5%, 25%, 90%) while keeping the other 7 languages equally weighted. So, if the selected language’s percentage is 12.5%, all 8 languages are equally weighted. We only use 4 languages (Tamil, Chinese, Icelandic, and English) as our selected languages and change the percentage of each of them. The *dependent variable* is the same as before – translation performance (spBLEU/chrF) on Flores101 (Goyal et al., 2021) devtest sets. We also examine the two *intermediate features*: closeness to the character level and UNK rate.

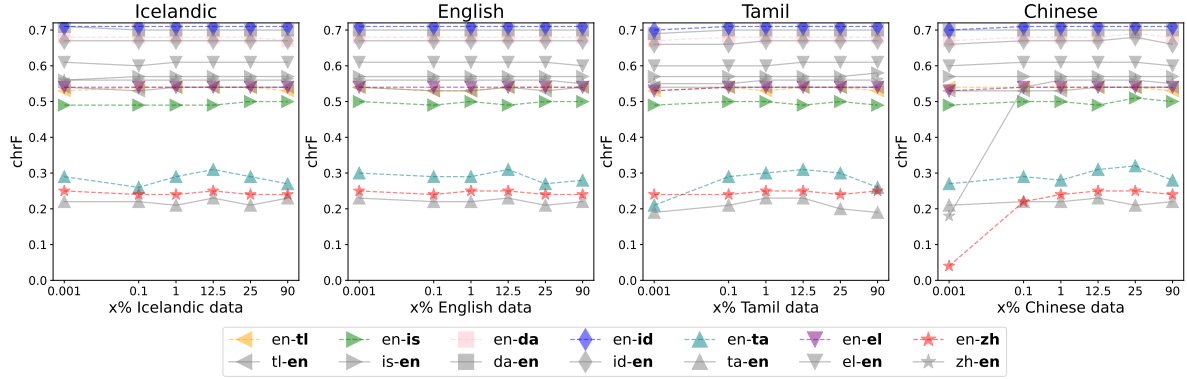


Figure 7.8: Translation results (chrF) of our main multilingual experiments. Markers have the same meanings as Figure 7.7. X axes are in log10 scale.

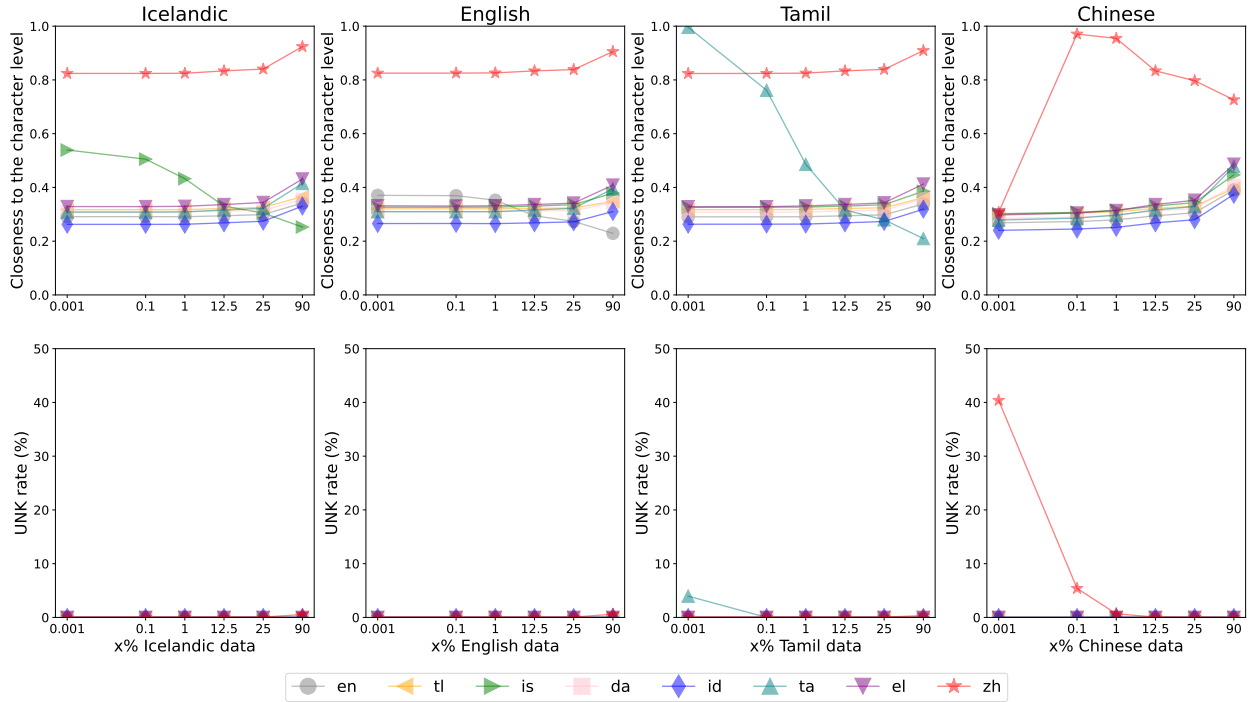


Figure 7.9: Intermediate features of our main multilingual experiments. Different from Figure 7.7, here, marker shapes and colors both denote the language. E.g., \triangle (ta) denotes Tamil features. X axes are in log10 scale.

7.4.2 Results & Ablations

Figure 7.7 illustrates the translation performance evaluated by spBLEU (chrF in Figure 7.8 shares the same trends). Figure 7.9 shows the features.

NMT performance is still quite robust to language imbalance especially when languages share scripts. As shown in Figure 7.7, for the two Latin languages (Icelandic and English), varying their percentages almost does not affect the performances. It is expectable for English because it can “never” be starved. But Icelandic’s performance drops at Icelandic=0.001% (English=99.999%) in bilingual experiments. We think it is because the involvement of multiple languages makes every language relatively less frequent, so the data ratio between Icelandic and any other language is not as disparate as 0.001:99.999 ($\approx 1:10^5$). This is also reflected by the trivial UNKs of all languages in Figure 7.9. For the two non-Latin languages (Tamil and Chinese), first, varying their percentages affects their own performances greatly while the performances of other languages still stay stable. And, their own performances drop quickly below 12.5% while dropping slower when percentages $\geq 12.5\%$.

Better performance is also often observed when languages are more balanced. In Figure 7.7, if we only consider the translation directions with great performance changes, i.e., Tamil and Chinese, they have relatively better performances around 12.5% when languages are balanced. We define *data ratio* as the lowest percentage of any language versus the highest percentage. So, the data ratio is 1 when the selected language’s percentage is 12.5%; while the data ratio is 0.07 when the selected language’s percentage is 1% ($\frac{0.01}{(1-0.01)/7} = 0.07$). Then, we compute the correlation between spBLEU scores and data ratios for each of the 4 selected languages. The average correlation is 0.49 (moderate correlation (Cohen, 1988)), which is consistent with what we observe in bilingual experiments.

Performance can drop without surpassing the thresholds of the two features. For Chinese, a more obvious performance drop happens at 0.1% following the indication of two features (UNK rate=5.4% and closeness to the character level=0.97). However, for Tamil, though its perfor-

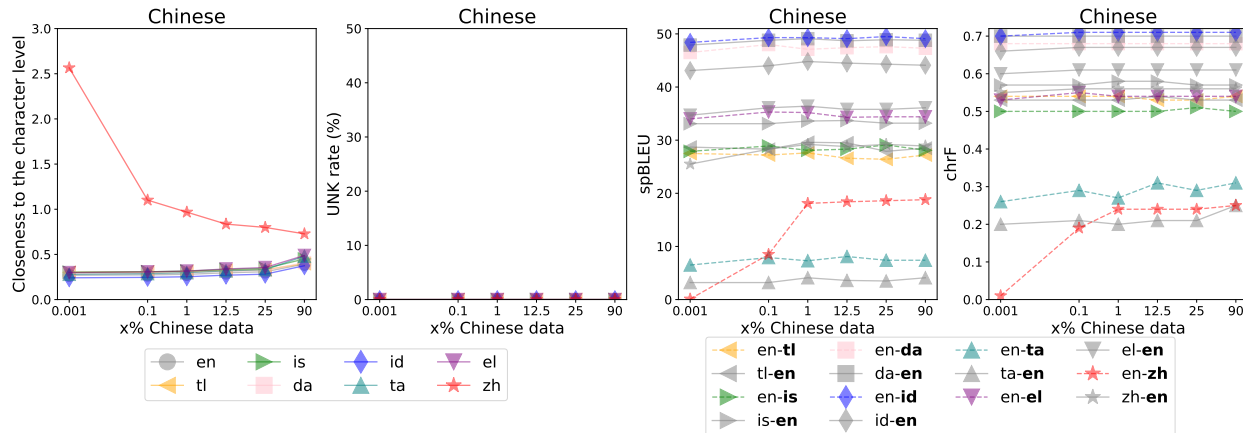


Figure 7.10: Intermediate features and translation results of the multilingual experiments with byte-fallback. Markers of the first two subplots have the same meanings as Figure 7.9, and markers of the second two subplots have the same meanings as Figure 7.7. X axes are in log10 scale.

mance drops at 1%, it has a trivial UNK rate and not long sentence length. This is probably due to the greatly compressed model capacity for each language pair, compared to bilingual experiments. Hence, though surpassing the thresholds can often hint at poor performances, it is neither a sufficient nor necessary condition.

Using byte-fallback still does not improve the robustness We apply *byte-fallback* under the setting of using Chinese as the selected language, and results are shown in Figure 7.10. Compared to Figure 7.7, though we observe slightly more stable performance when Chinese $\geq 1\%$, the translation result drops more dramatically when Chinese $\leq 0.1\%$.

NMT is more sensitive to language imbalance in model training. In both bilingual or multilingual settings, we find that the performance is quite robust to language imbalance and relatively better performance is often observed when languages are more balanced. In other words, we want to set sampling factor $S = 0$, following the temperature sampling paradigm (Devlin et al., 2019). However, many existing works show significantly different performances of different S , and the best S is around 0.2 to 0.7 (Arivazhagan et al., 2019; Conneau and Lample, 2019; Xue et al., 2021b). We think this inconsistency has resulted from the fact that we fix $S = 0.2$ for model training while only varying it (via changing data percentages) for tokenizer training. We conjecture that NMT is more sensitive to language imbalance in model training. To verify this,

	S	*-en								en-*								overall
		tl	is	da	id	ta	el	zh	avg.	tl	is	da	id	ta	el	zh	avg.	avg.
tokenizer	0	29.6	33.1	48.5	44.6	4.0	36.1	29.1	32.1	27.9	27.4	47.0	49.1	8.7	34.6	18.5	30.5	31.3
	0.3	28.6	33.6	49.0	44.0	3.4	36.6	28.5	32.0	26.6	27.5	46.2	48.7	7.6	34.2	18.4	29.9	30.9
	1	29.0	32.4	48.4	44.1	3.4	35.6	28.8	31.7	27.5	29.0	47.8	49.7	7.6	34.6	19.1	30.8	31.2
model	0	28.2	32.6	47.4	41.6	3.6	34.2	26.7	30.6	26.9	27.8	45.9	47.3	6.9	33.1	17.1	28.3	29.9
	0.2	29.6	33.1	48.5	44.6	4.0	36.1	29.1	32.1	27.9	27.4	47.0	49.1	8.7	34.6	18.5	30.5	31.3
	1	27.2	33.3	49.7	46.2	4.0	37.6	31.7	32.9	16.9	25.7	47.8	50.1	3.4	35.7	19.8	28.5	30.7

Table 7.3: Comparison of language sampling factors used in tokenizer or model training. All numbers are spBLEU. S is the exponential factor used in temperature sampling (see Section 7.2.2).

first, we fix model training sampling $S = 0.2$ and compare 3 tokenizer training sampling factors ($S = 0, 0.3, 1.0$). Results are shown in the second row (starting with “tokenizer”) in Table 7.3. Though with small differences (0.4, 0.1 points), $S = 0$ overall works best. Second, we fix tokenizer training sampling $S = 0$ and compare 3 model training sampling factors ($S = 0, 0.2, 1.0$). As shown in Table 7.3, the differences are more prominent (1.4, 0.6 points), and $S = 0.2$ overall works best. Hence, for tokenizer training, we want languages to be balanced, whereas, for model training, we want to flatten the original distribution to some degree but not to uniform distribution. And we want to pay more attention to sampling for model training because NMT is more sensitive to it.

7.5 Implementation Details

We implement translation models using fairseq.⁸ During training, we use Adam optimizer (Kingma and Ba, 2015), learning rate=0.001, and warmup for 2 epochs. We use batch size=4K tokens and gradient accumulation=4. For bilingual experiments, we use 8 NVIDIA Tesla V100 Volta GPUs for each experiment, and we run 3 seeds (2, 7, 42) for each experiment and report the average. For multilingual experiments, we use 64 GPUs and only run seed=2 for each experiment. We apply early stop with patience of 20 epochs. During testing, we use batch size=32 sentences and beam size=5.

⁸<https://github.com/pytorch/fairseq>

7.6 Conclusion

We systematically analyze how language imbalance in multilingual tokenizer training affects translation performances. Overall, we find that NMT performance is quite robust to language imbalance especially when languages share scripts. Better performance is often achieved when languages are more balanced. We suggest keeping the involved languages as balanced as possible in the tokenizer training corpus and evaluating pretrained tokenizers on an evaluation set to make sure no language's UNK rate \geq around 3.7% and no language's closeness to the character level \geq around 0.87. We hope our work can provide some guidance for future multilingual tokenizer training and usage.

x% English	English	Indonesian
0	_ " We _ now _ have _ 4 - mon th - ol d _ mi ce _ th at _ a re _ non - dia be tic _ th at _ us ed _ to _ be _ dia be tic , " _ he _ ad de d .	_ " S a at _ ini _ ada _ men ci t _ umur _ 4 _ bulan _ non dia bet es _ yang _ dulu nya _ diabetes , " _ tamba h nya .
0.1	_ " We _ now _ have _ 4 - mon th - ol d _ mi ce _ th at _ a re _ non - dia be tic _ th at _ us ed _ to _ be _ dia be tic , " _ he _ ad de d .	_ " S a at _ ini _ ada _ men ci t _ umur _ 4 _ bulan _ non dia bet es _ yang _ dulu nya _ diabetes , " _ tamba h nya .
50	_ " We _ now _ have _ 4 - mon th - old _ mi ce _ that _ are _ non - dia be tic _ that _ used _ to _ be _ dia be tic , " _ he _ added .	_ " S a at _ ini _ ada _ men ci t _ umur _ 4 _ bulan _ non dia bet es _ yang _ dulu nya _ diabetes , " _ tamba h nya .
99.9	_ " We _ now _ have _ 4 - mon th - old _ mi ce _ that _ are _ non - dia be tic _ that _ used _ to _ be _ di a be tic , " _ he _ added .	_ " S a at _ in i _ a da _ men ci t _ um ur _ 4 _ bu lan _ non di ab et es _ ya ng _ du lu nya _ diabetes , " _ ta mb ah nya .
100	_ " We _ now _ have _ 4 - mon th - old _ mi ce _ that _ are _ non - dia be tic _ that _ used _ to _ be _ di a be tic , " _ he _ added .	_ " S a at _ in i _ a da _ men ci t _ um ur _ 4 _ b ul an _ non di ab et es _ ya ng _ du lu ny a _ diabetes , " _ ta mb ah nya .
x% English	English	Chinese
0	_ " We _ now _ have _ 4 - mon th - ol d _ mi ce _ th at _ are _ non - dia be tic _ th at _ us ed _ to _ be _ dia be tic , " _ he _ ad de d .	_ 他补充道: “我们现在有 _ 4 _ 个月大没有 糖尿病的老鼠, 但它们曾经得过该病。”
0.1	_ " We _ now _ have _ 4 - mon th - ol d _ mi ce _ th at _ are _ non - dia be tic _ th at _ us ed _ to _ be _ dia be tic , " _ h e _ ad de d .	_ 他补充道: “我们现在有 _ 4 _ 个月大没有 糖尿病的老鼠, 但它们曾经得过该病。”
50	_ " We _ now _ have _ 4 - mon th - old _ mi ce _ that _ are _ no n - dia be tic _ that _ used _ to _ be _ dia be tic , " _ he _ add ed .	_ 他补充道: “我们现在有 _ 4 _ 个月大没有 糖尿病的老鼠, 但它们曾经得过该病。”
99.9	_ " We _ now _ have _ 4 - mon th - old _ m ice _ that _ are _ non - dia be tic _ that _ used _ to _ be _ dia be tic , " _ he _ added .	_ <unk> : “<unk> _ 4 _ <unk> , <unk> ”
100	_ " We _ now _ have _ 4 - mon th - old _ m ice _ that _ are _ non - dia be tic _ that _ used _ to _ be _ dia be tic , " _ he _ added .	_ <unk> : “<unk> _ 4 _ <unk> , <unk> ”

Table 7.4: Examples of how sentences in English, Indonesian, and Chinese are tokenized at different English percentages under our main bilingual setting (Section 7.3.1). The sentence is the first sentence of Flores101 devtest set. Subwords are separated by whitespaces, and unknown tokens are replaced by ‘<unk>’.

CHAPTER 8: SUMMARY, LIMITATIONS, ETHICS, AND FUTURE WORK

8.1 Summary of Contributions

This thesis focuses on improving the *reliability* or *inclusiveness* of natural language generation (NLG). The main contributions can be summarized into the following three aspects.

Alternative learning objective for training more reliable NLG models. We point out that the typical learning objective, maximum likelihood estimation (MLE), used for training NLG models is not always sufficient for training a reliable model. For example, in Question Generation (QG), MLE does not explicitly reflect the requirement that generated questions should be answerable by the given answer. Therefore, the output questions are often unanswerable by the answer. In (Zhang and Bansal, 2019) (Chapter 2), we proposed to use an external pretrained QA model to verify the answerability of the generated question and use it as a reward to train the QG model. We showed that our method greatly improved question generation performance and the answerability of questions. For language modeling, we found that MLE-trained LMs tend to over-generalize, in the sense of having larger support than human LM distribution and thus producing non-human-like text when random sampling from the model. In our recent work (Zhang et al., 2023b) introduced in Chapter 3, we proposed a novel MixCE training objective that minimizes a mixture of forward cross-entropy (CE) (which is equivalent to MLE) and reverse CE. We demonstrated that MixCE effectively alleviates the over-generalization problem of MLE and leads to better LM performance.

More reliable summary evaluation methods. How to reliably evaluate model-generated text is a long-standing problem in the NLG literature. Different NLG tasks have different evaluation methodologies, though they are usually based on similar ideas. In this thesis, we particularly fo-

cus on one of the NLG tasks, text summarization, and propose metrics, benchmarks, and protocols to improve the reliability of the summary evaluation. Usually, human evaluation is viewed as the gold standard evaluation method of NLG. However, it is expensive, time-consuming, and non-reproducible. In contrast, automatic evaluation metrics are low-cost, fast, and reproducible, yet it is often poorly correlated with human judgment. Therefore, in (Zhang and Bansal, 2021) (Chapter 4), we proposed a method to combine human and automatic evaluations to achieve semi-automatic summary evaluation. We showed that our approach can find a good trade-off between both worlds: being cheap, fast, and reproducible while being more correlated with human judgment than other existing automatic metrics. Recently, an increasing number of works have been studying the faithfulness (or factually consistency) issues of text summarization because models frequently hallucinate new information or change the meaning of the source (Cao et al., 2018; Maynez et al., 2020). However, these works have only focused on abstractive summarization (generating novel sentences) rather than extractive summarization (extracting sentences from the source). Even though extractive models are more reliable in terms of faithfulness, in our recent work (Zhang et al., 2023a) (Chapter 5), we showed that there are a non-trivial number of unfaithfulness problems existing in extracted summaries produced by state-of-the-art extractive systems, and we proposed a new metric to better detect these unfaithful extractive summaries.

NLG for endangered or low-resource languages. There are over 6,500 languages spoken or signed in the world today. However, only a handful of languages are systematically represented in NLG (or in general NLP) technologies (Joshi et al., 2020b). To support as many languages as possible is an important and meaningful mission of the whole NLP community. This thesis has worked on the language processing of an endangered Native American Language, Cherokee. As discussed in Chapter 6, we collected a Cherokee-English parallel dataset (Zhang et al., 2020b) and developed the first set of Cherokee-English translation systems and an online demo (Zhang et al., 2021b) (<https://chren.cs.unc.edu/>). This demo has been tested and used by Cherokee speakers and learners. The dataset is used by the machine translation assignment of the Stanford CS224n NLP course, and the demo was featured by UNC Research in a news article. Be-

sides machine translation, we also introduced a more complete roadmap for using NLP to help revitalize endangered languages like Cherokee (Zhang et al., 2022b), in which we proposed suggestions to NLP practitioners, approaches of NLP-assisted language education, and directions for Cherokee language processing. On the other hand, tokenization is a necessary processing step of most NLG models. When multiple languages are involved, usually one multilingual tokenizer is trained. However, due to the different amounts of data from different languages, low-resource languages may not be well represented in the learned vocabulary. In our work discussed in Chapter 7 (Zhang et al., 2022a), we studied how language imbalance in tokenization affects the performance of multilingual translation. We found that translation models are surprisingly robust to language imbalance, nonetheless, better performance is often observed when languages are more balanced. We provided best practices for training and using multilingual tokenizers.

8.2 Limitations and Future Work

In an idealized setting, with unlimited training data and model capacity, as well as a perfect optimizer, fitting Q_θ with MLE will learn a distribution as close to P as we like. In other words, when a large amount of clean data is used, the over-generalization problem caused by MLE is less noticeable, just like how we see models trained with large-scale data usually have better performance. And thus, alternative learning objectives will become less useful in these settings. The novel learning objective components we introduced, QAP reward for QG and reverse CE for LM, can be easily “gamed” by the model, e.g., copying the answer to the generated question, always outputting one single piece of human-like text. Therefore, it is critical to mix them with MLE. However, we found that the best mixing ratio is different across different settings. Our current best practice is to tune the mixing ratio on a development set, but it is less obvious how to use these learning objectives during pretraining when it is too expensive to tune hyperparameters. Therefore, how to find a universal mixing ratio or how to determine it automatically is an important problem to resolve in the future.

Our semi-automatic summary evaluation is based on the reference-based evaluation protocol, Pyramid (Nenkova and Passonneau, 2004) and LitePyramid (Shapira et al., 2019). However, a lot of people criticize the low quality of reference summaries in summarization datasets (Bommasani and Cardie, 2020). In general, reference-based evaluations have more controllability and reproducibility while their evaluation capacity is upper bounded by the references. In contrast, reference-free evaluations are more difficult to control and less reproducible but more flexible. How to resolve this dilemma is an interesting future work.

The conclusions of our “extractive is not faithful” work will be more useful for summarization tasks where extractive methods perform decently well compared to extremely abstractive summarization tasks. Experts conducted our data annotations; hence, to scale up data annotation by working with crowdsourcing workers may require additional training for the workers. Our ExtEval metric is designed for extractive summarization, which is currently not directly applicable for abstractive summaries except for SentiBias. As our data is collected on CNN/DM, the percentages of each error type may change when evaluating a different summarization dataset, though we believe that the conclusion, extractive is not faithful, will not change.

In the series of Cherokee-related works, we are inspired by the practice of Cherokee Language Revitalization. Our conclusions and suggestions may or may not generalize to other endangered languages. For example, since Cherokee has its own syllabary and can be written down, we are interested in speech recognition for audio transcription. However, some oral languages may want to prioritize translation over transcription to tackle the transcription bottleneck (Bird, 2020b). In addition, our position is influenced by Crystal (2014), who thinks using electronic technology is important for language revitalization. Therefore, a lot of our proposals may have an assumption that computers and the Internet have been or can be widely accepted and used in the indigenous community. However, it may not be true in every indigenous community.

Lastly, our analysis of language imbalance in multilingual tokenization is an empirical study. Our observations and conclusions are made based on our experiments, which may or may not be generalizable to other settings. We tried our best to include diverse languages, but still, our exper-

iments are English-centric and at most have 8 languages involved. We tended to believe that our main observations are generalizable to other experimental settings, while the exact thresholds of the two features (UNK rate and closeness to the character level) for indicating poor downstream performance may not always hold.

8.3 Ethical Considerations

Despite the impressive progress of NLG and the effort made by this thesis, NLG models often do not distinguish fact from fiction, so they can not support use cases that require the generated text to be true. Additionally, models reflect the biases inherent to the data they were trained on, so they can not be deployed into systems that interact with humans unless the deployers first carry out a study of biases relevant to the intended use case.

Automatic and semi-automatic NLG evaluation metrics are inherently biased by how they are designed and thus can not replace human evaluation. Nonetheless, human evaluation is also not always trustworthy because it is also biased or limited by its evaluation protocol and how human evaluators are instructed. Therefore, multiple and diverse evaluations should be applied in quality-sensitive scenarios and evaluations should be carried out according to the intended use case.

The ethical foundation of working with indigenous people, e.g., native speakers from the Cherokee community, has been addressed in Section 6.5.1. To summarize, as NLP practitioners, who are usually “outsiders” of indigenous communities, we need to keep in mind their basic need: the need for respect, reciprocity, and understanding. We need to weigh the burden we put upon the native speakers against the benefit that the research can bring back to their community. And lastly, we need to decolonize our research and form a sustainable collaboration community with them.

The main ethical concern of the multilingual tokenization work is that we have many experiments and it is not very easy to finish them without a decent number of computation resources. In total, we have 1890 bilingual experiments. Each experiment takes from less than 1 hour to

about 2 days (based on the training data size) using 8 NVIDIA Tesla V100 Volta GPUs. And, we have 31 multilingual experiments in total, and each experiment takes 1.5 days using 64 GPUs. However, we expect that our empirical results can help guide the training and usage of multilingual tokenizers, so future works do not have to re-conduct these expensive investigations.

REFERENCES

- (2008). *Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008*. NIST.
- (2009). *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.
- Albee, E. (2017). Immersion schools and language learning: A review of cherokee language revitalization efforts among the eastern band of cherokee indians.
- Alberti, C., Andor, D., Pitler, E., Devlin, J., and Collins, M. (2019). Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.
- Amrhein, C. and Sennrich, R. (2021). How suitable are subword segmentation strategies for translating non-concatenative morphology? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 689–705, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., et al. (2019). Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.
- Austin, P. K. and Sallabank, J. (2011). *The Cambridge handbook of endangered languages*. Cambridge University Press.
- Baevski, A. and Auli, M. (2019). Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.
- Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Barzilay, R., McKeown, K. R., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA. Association for Computational Linguistics.

- Basu, S., Ramachandran, G. S., Keskar, N. S., and Varshney, L. R. (2021). {MIROSTAT}: A {neural} {text} {decoding} {algorithm} {that} {directly} {controls} {perplexity}. In *International Conference on Learning Representations*.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.
- Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.
- Bhandari, M., Gour, P. N., Ashfaq, A., Liu, P., and Neubig, G. (2020). Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.
- Bird, S. (2020a). Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Bird, S. (2020b). Sparse transcription. *Computational Linguistics*, 46(4):713–744.
- Bird, S. (2021). Lt4all!? rethinking the agenda. In *EMNLP*.
- Blasi, D., Anastasopoulos, A., and Neubig, G. (2022). Systematic inequalities in language technology performance across the world’s languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505, Dublin, Ireland. Association for Computational Linguistics.
- Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.
- Bommasani, R. and Cardie, C. (2020). Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.
- Bostrom, K. and Durrett, G. (2020). Byte pair encoding is suboptimal for language model pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4617–4624.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

- Britannica (2021). Sequoyah. *Encyclopedia Britannica*, 28 Jul.
- Brown, A. L. and Day, J. D. (1983). Macrorules for summarizing texts: The development of expertise. *Journal of verbal learning and verbal behavior*, 22(1):1–14.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS’20, Red Hook, NY, USA. Curran Associates Inc.
- Bryant, C., Felice, M., Andersen, Ø. E., and Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Bustamante, G., Oncevay, A., and Zariquiey, R. (2020). No data to crawl? monolingual corpus creation from PDF files of truly low-resource languages in Peru. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2914–2923, Marseille, France. European Language Resources Association.
- Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. (2020). Language gans falling short. In *International Conference on Learning Representations*.
- Cao, S. and Wang, L. (2021). CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Cao, Z., Wei, F., Li, W., and Li, S. (2018). Faithful to the original: Fact aware neural abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Carreras, X. and Màrquez, L. (2005). Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)*, pages 152–164.
- Chen, J., Fife, J. H., Bejar, I. I., and Rupp, A. A. (2016). Building e-rater® scoring models using machine learning methods. *ETS Research Report Series*, 2016(1):1–12.
- Chen, S., Zhang, F., Sone, K., and Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Chen, Y.-C. and Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.
- Cheung, J. C. (2008). Comparing abstractive and extractive summarization of evaluative text: controversiality and content selection. *B. Sc.(Hons.) Thesis in the Department of Computer Science of the Faculty of Science, University of British Columbia*, 47.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Christodouloupoulos, C. and Steedman, M. (2015). A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49(2):375–395.
- Chung, H. W., Garrette, D., Tan, K. C., and Riesa, J. (2020). Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546.
- Chung, J., Cho, K., and Bengio, Y. (2016). 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.
- Clark, C. and Gardner, M. (2018). Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855.
- Clark, E., Celikyilmaz, A., and Smith, N. A. (2019). Sentence mover’s similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760.
- Clark, J. H., Garrette, D., Turc, I., and Wieting, J. (2022). Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Routledge.
- Cohn, T. and Lapata, M. (2008). Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK. Coling 2008 Organizing Committee.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

- Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.
- Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.
- Crystal, D. (2014). *Language Death*. Canto Classics. Cambridge University Press.
- Dagan, I., Glickman, O., and Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.
- Datta, R. (2018). Decolonizing both researcher and research and its effectiveness in indigenous research. *Research Ethics*, 14(2):1–24.
- Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*.
- Deutsch, D., Bedrax-Weiss, T., and Roth, D. (2021). Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.
- Deutsch, D. and Roth, D. (2020). SacreROUGE: An open-source library for using and developing summarization evaluation metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. Association for Computational Linguistics.
- Deutsch, D. and Roth, D. (2021). Understanding the extent to which summarization evaluation metrics measure the information quality of summaries. *CoNLL*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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. Association for Computational Linguistics.
- Dhingra, B., Danish, D., and Rajagopal, D. (2018). Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587.
- Ding, S., Renduchintala, A., and Duh, K. (2019). A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.
- Dobrin, L. M., Austin, P. K., and Nathan, D. (2007). Dying to be counted: The commodification of endangered languages in documentary linguistics. In *Proceedings of the conference on language documentation and linguistic theory*, pages 59–68. SOAS London.

- Domingo, M., Garcia-Martinez, M., Helle, A., Casacuberta, F., and Herranz, M. (2018). How much does tokenization affect neural machine translation? *arXiv preprint arXiv:1812.08621*.
- Dong, L., Mallinson, J., Reddy, S., and Lapata, M. (2017). Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.
- Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.
- Dong, Y., Shen, Y., Crawford, E., van Hoof, H., and Cheung, J. C. K. (2018). BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.
- Dreyer, M., Liu, M., Nan, F., Atluri, S., and Ravi, S. (2021). Analyzing the abstractiveness-factuality tradeoff with nonlinear abstractiveness constraints. *arXiv preprint arXiv:2108.02859*.
- Du, L., Hennigen, L. T., Pimentel, T., Meister, C., Eisner, J., and Cotterell, R. (2022). A measure-theoretic characterization of tight language models. *arXiv preprint arXiv:2212.10502*.
- Du, X. and Cardie, C. (2018). Harvesting paragraph-level question-answer pairs from wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917.
- Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.
- Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.
- Durmus, E., He, H., and Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.
- Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.
- Eikema, B. and Aziz, W. (2020). Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

- Eyal, M., Baumel, T., and Elhadad, M. (2019). Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948.
- Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. (2021). SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., and Gurevych, I. (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.
- Fan, A., Lewis, M., and Dauphin, Y. (2018a). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Fan, L., Yu, D., and Wang, L. (2018b). Robust neural abstractive summarization systems and evaluation against adversarial information. In *Workshop on Interpretability and Robustness in Audio, Speech, and Language (IRASL)*. Neural Information Processing Systems.
- Feeling, D. (1975). *Cherokee-English Dictionary*. Cherokee Nation of Oklahoma.
- Feeling, D. (1994). *A structured approach to learning the basic inflections of the Cherokee verb*. Indian University Press, Bacone College.
- Feeling, D. (2018). *Cherokee Narratives: A Linguistic Study*. University of Oklahoma Press.
- Firat, O., Cho, K., and Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875.
- Fischer, T. (2021). Finding factual inconsistencies in abstractive summaries. Master’s thesis, Universität Hamburg.
- Fomicheva, M., Sun, S., Yankovskaya, L., Blain, F., Chaudhary, V., Fishel, M., Guzmán, F., and Specia, L. (2020a). BERGAMOT-LATTE submissions for the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.
- Fomicheva, M., Sun, S., Yankovskaya, L., Blain, F., Guzmán, F., Fishel, M., Aletras, N., Chaudhary, V., and Specia, L. (2020b). Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.
- Frey, B. E. (2013). *Toward a general theory of language shift: A case study in Wisconsin German and North Carolina Cherokee*. PhD thesis, The University of Wisconsin-Madison.

- Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.-P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., and Darrell, T. (2018). Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325.
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.
- Gabriel, S., Celikyilmaz, A., Jha, R., Choi, Y., and Gao, J. (2021). GO FIGURE: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 478–487, Online. Association for Computational Linguistics.
- Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.
- Gao, T., Yao, X., and Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Gao, Y., Sun, C., and Passonneau, R. J. (2019). Automated pyramid summarization evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 404–418.
- Gao, Y., Zhao, W., and Eger, S. (2020). Supert: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354.
- Garcia, I. (2013). Learning a language for free while translating the web. does duolingo work? *International Journal of English Linguistics*, 3(1):19.
- Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.
- Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.
- Gass, S. M. and Mackey, A. (2013). *The Routledge handbook of second language acquisition*. Routledge.
- Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. (2020). RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

- Gehrmann, S., Adewumi, T., Aggarwal, K., Ammanamanchi, P. S., Aremu, A., Bosselut, A., Chandu, K. R., Clinciu, M.-A., Das, D., Dhole, K., Du, W., Durmus, E., Dušek, O., Emezue, C. C., Gangal, V., Garbacea, C., Hashimoto, T., Hou, Y., Jernite, Y., Jhamtani, H., Ji, Y., Jolly, S., Kale, M., Kumar, D., Ladhak, F., Madaan, A., Maddela, M., Mahajan, K., Mahamood, S., Majumder, B. P., Martins, P. H., McMillan-Major, A., Mille, S., van Miltenburg, E., Nadeem, M., Narayan, S., Nikolaev, V., Niyongabo Rubungo, A., Osei, S., Parikh, A., Perez-Beltrachini, L., Rao, N. R., Raunak, V., Rodriguez, J. D., Santhanam, S., Sedoc, J., Sellam, T., Shaikh, S., Shimorina, A., Sobrevilla Cabezudo, M. A., Strobel, H., Subramani, N., Xu, W., Yang, D., Yerukola, A., and Zhou, J. (2021). The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.
- Gentzkow, M., Shapiro, J. M., and Stone, D. F. (2015). Media bias in the marketplace: Theory. In *Handbook of media economics*, volume 1, pages 623–645. Elsevier.
- Gerz, D., Vulić, I., Ponti, E. M., Reichart, R., and Korhonen, A. (2018). On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327.
- Giannakopoulos, G. and Karkaletsis, V. (2011). Autosummeng and memog in evaluating guided summaries. In *TAC*.
- Giannakopoulos, G., Karkaletsis, V., Vouros, G., and Stamatopoulos, P. (2008). Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):1–39.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Gowda, T. and May, J. (2020). Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.
- Goyal, N., Gao, C., Chaudhary, V., Chen, P., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. (2021). The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.
- Goyal, T. and Durrett, G. (2020). Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.
- Goyal, T. and Durrett, G. (2021). Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

- Goyal, T., Li, J. J., and Durrett, G. (2022). Snac: Coherence error detection for narrative summarization. *arXiv preprint arXiv:2205.09641*.
- Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.
- Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.
- Gulick, J. (1958). Language and passive resistance among the eastern cherokees. *Ethnohistory*, 5(1):60–81.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.
- Gupta, V. and Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268.
- Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. (2019). The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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 6098–6111, Hong Kong, China. Association for Computational Linguistics.
- Hashimoto, T. B., Zhang, H., and Liang, P. (2019). Unifying human and statistical evaluation for natural language generation. 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 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.
- He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., and Ma, W.-Y. (2016). Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828.
- He, L., Lee, K., Lewis, M., and Zettlemoyer, L. (2017). Deep semantic role labeling: What works and what’s next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.
- Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria. Association for Computational Linguistics.
- Heilman, M. and Smith, N. A. (2010). Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American*

- Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics.
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *NeurIPS*, pages 1693–1701.
- Hewitt, J., Manning, C. D., and Liang, P. (2022). Truncation sampling as language model desmoothing. In *Findings of the Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*.
- Higgins, J. (1983). Computer assisted language learning. *Language Teaching*, 16(2):102–114.
- Hill, J. and Simha, R. (2016). 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, San Diego, CA. Association for Computational Linguistics.
- Hirao, T., Kamigaito, H., and Nagata, M. (2018). Automatic pyramid evaluation exploiting EDU-based extractive reference summaries. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4186, Brussels, Belgium. Association for Computational Linguistics.
- Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Hosking, T. and Riedel, S. (2019). Evaluating rewards for question generation models. *arXiv preprint arXiv:1902.11049*.
- Hu, Y., Zhang, S., Sathy, V., Panter, A., and Bansal, M. (2022). SETSum: Summarization and visualization of student evaluations of teaching. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pages 71–89, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.
- Huszár, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.
- Irsoy, O. (2019). On expected accuracy. *arXiv preprint arXiv:1905.00448*.
- Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015). Montreal neural machine translation systems for WMT’15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal. Association for Computational Linguistics.

- Ji, H., Ke, P., Hu, Z., Zhang, R., and Huang, M. (2023). Tailoring language generation models under total variation distance. In *The Eleventh International Conference on Learning Representations*.
- Johnson, M., Schuster, M., Le, Q., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020a). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020b). 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. Association for Computational Linguistics.
- Joshi, V., Peters, M., and Hopkins, M. (2018). Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199, Melbourne, Australia. Association for Computational Linguistics.
- Julian, C. (2010). *A history of the Iroquoian languages*. PhD thesis, University of Manitoba.
- Kang, D. and Hashimoto, T. B. (2020). Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.
- Kim, Y., Dyer, C., and Rush, A. (2019). Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.
- Kim, Y., Lee, H., Shin, J., and Jung, K. (2018). Improving neural question generation using answer separation. *arXiv preprint arXiv:1809.02393*.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kintsch, W. and van Dijk, T. (1978). Cognitive psychology and discourse: Recalling and summarizing stories. *Current trends in text linguistics*, pages 61–80.
- Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

- Kocmi, T. and Bojar, O. (2018). Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. *ACL 2017*, page 28.
- Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.
- Krippendorff, K. (2011). Computing krippendorff’s alpha-reliability. *Computing*, 1.
- Krishna, K., Chang, Y., Wieting, J., and Iyyer, M. (2022). Rankgen: Improving text generation with large ranking models. In *Empirical Methods in Natural Language Processing*.
- Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., and Socher, R. (2019). Neural text summarization: A critical evaluation. 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 540–551.
- Kryscinski, W., McCann, B., Xiong, C., and Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.
- Kudo, T. and Richardson, J. (2018). 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.
- Kumar, V., Ramakrishnan, G., and Li, Y.-F. (2018). A framework for automatic question generation from text using deep reinforcement learning. *arXiv preprint arXiv:1808.04961*.
- Labutov, I., Basu, S., and Vanderwende, L. (2015). Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 889–898.

- Ladhak, F., Durmus, E., He, H., Cardie, C., and McKeown, K. (2022). Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421, Dublin, Ireland. Association for Computational Linguistics.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6405–6416.
- Lambert, P., Schwenk, H., Servan, C., and Abdul-Rauf, S. (2011). Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 284–293, Edinburgh, Scotland. Association for Computational Linguistics.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.
- Lee, J., Cho, K., and Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.
- Lewis, M. and Fan, A. (2019). Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations (ICLR)*.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020a). 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.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020b). 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. Association for Computational Linguistics.
- Li, H., Einolghozati, A., Iyer, S., Paranjape, B., Mehdad, Y., Gupta, S., and Ghazvininejad, M. (2021). EASE: Extractive-abstractive summarization end-to-end using the information bottleneck principle. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 85–95, Online and in Dominican Republic. Association for Computational Linguistics.

- Li, H., Zhu, J., Zhang, J., and Zong, C. (2018). Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Li, J., Monroe, W., and Jurafsky, D. (2016). A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.
- Li, M., Roller, S., Kulikov, I., Welleck, S., Boureau, Y.-L., Cho, K., and Weston, J. (2020). Don’t say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.
- Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., and Lewis, M. (2022). Contrastive decoding: Open-ended text generation as optimization.
- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Lin, C.-Y., Cao, G., Gao, J., and Nie, J.-Y. (2006). An information-theoretic approach to automatic evaluation of summaries. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 463–470, New York City, USA. Association for Computational Linguistics.
- Lindberg, D., Popowich, F., Nesbit, J., and Winne, P. (2013). Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.
- Liu, B., Zhao, M., Niu, D., Lai, K., He, Y., Wei, H., and Xu, Y. (2019a). Learning to generate questions by learning what not to generate. *arXiv preprint arXiv:1902.10418*.
- Liu, T., Wang, K., Sha, L., Chang, B., and Sui, Z. (2018). Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. 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 3721–3731.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

- Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Màrquez, L., Recasens, M., and Sapena, E. (2013). Coreference resolution: An empirical study based on semeval-2010 shared task 1. *Lang. Resour. Eval.*, 47(3):661–694.
- Maybury, M. (1999). *Advances in automatic text summarization*. MIT press.
- Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Meister, C., Pimentel, T., Wiher, G., and Cotterell, R. (2022). Locally typical sampling. *Transactions of the Association for Computational Linguistics*, abs/2202.00666.
- Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2016). Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.
- Mikolov, T., Karafiát, M., Burget, L., et al. (2010). Recurrent neural network based language model. In *In INTERSPEECH 2010*,. Citeseer.
- Montgomery-Anderson, B. (2008). *A reference grammar of Oklahoma Cherokee*. PhD thesis, University of Kansas.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Nanba, H. and Okumura, M. (2000). Producing more readable extracts by revising them. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.
- Narayan, S., Cohen, S. B., and Lapata, M. (2018a). Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

- Narayan, S., Cohen, S. B., and Lapata, M. (2018b). Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.
- Nation, C. (2001). Ga-du-gi: A vision for working together to preserve the Cherokee language. report of a needs assessment survey and a 10-year language revitalization plan.
- NCAI (2020). *Tribal nations and the United States: An introduction*. National Congress of American Indians Washington, DC.
- Nema, P. and Khapra, M. M. (2018). Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959.
- Nenkova, A. and McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.
- Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- O’Connor, C. and Weatherall, J. O. (2019). The misinformation age. In *The Misinformation Age*. Yale University Press.
- Ott, M., Auli, M., Grangier, D., and Ranzato, M. (2018). Analyzing uncertainty in neural machine translation. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965. PMLR.

- Padmakumar, V. and He, H. (2021). Unsupervised extractive summarization using pointwise mutual information. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2505–2512, Online. Association for Computational Linguistics.
- Pagnoni, A., Balachandran, V., and Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.
- Palmer, M., Gildea, D., and Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.
- Pang, R. Y. and He, H. (2021). Text generation by learning from demonstrations. In *ICLR*.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*.
- Parikh, A., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., and Das, D. (2020). ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.
- Passonneau, R. J. (2010). Formal and functional assessment of the pyramid method for summary content evaluation. *Natural Language Engineering*, 16(2):107.
- Pasunuru, R. and Bansal, M. (2018). Multi-reward reinforced summarization with saliency and entailment. In *NAACL*.
- Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.
- Peake Raymond, M. (2008). The cherokee nation and its language: Tsalagi ayeli ale uniwon-ishisdi. *Tahlequah, OK: Cherokee Nation*.
- Perdue, T. and Green, M. D. (2007). *The Cherokee nation and the trail of tears*. Penguin.
- Peters, M. t. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL*.
- Peyrard, M., Botschen, T., and Gurevych, I. (2017). Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.
- Pillutla, K., Liu, L., Thickstun, J., Welleck, S., Swayamdipta, S., Zellers, R., Oh, S., Choi, Y., and Harchaoui, Z. (2022). Mauve scores for generative models: Theory and practice. *arXiv preprint arXiv:2212.14578*.

- Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., and Harchaoui, Z. (2021). Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.
- Popov, V. and Kudinov, M. (2018). Fine-tuning of language models with discriminator. *arXiv preprint arXiv:1811.04623*.
- Popović, M. (2015). 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. Association for Computational Linguistics.
- Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.
- Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Pratt, R. H. (2013). 35. *The Advantages of Mingling Indians with Whites*. Harvard University Press.
- Press, O. and Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.
- Prud’hommeaux, E., Jimerson, R., Hatcher, R., and Michelson, K. (2021). Automatic speech recognition for supporting endangered language documentation. *Language Documentation & Conservation*, 15:491–513.
- Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.
- Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195. IEEE.
- Rigutini, L., Diligenti, M., Maggini, M., and Gori, M. (2012). Automatic generation of crossword puzzles. *International Journal on Artificial Intelligence Tools*, 21(03):1250014.
- Rijhwani, S., Anastasopoulos, A., and Neubig, G. (2020). OCR Post Correction for Endangered Language Texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942, Online. Association for Computational Linguistics.
- Ruan, Q., Ostendorff, M., and Rehm, G. (2022). HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics.
- Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., and Gurevych, I. (2021). How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135.
- Sachan, M. and Xing, E. (2018). Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640.
- Saggion, H. and Poibeau, T. (2013). Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer.
- Salesky, E., Etter, D., and Post, M. (2021). Robust open-vocabulary translation from visual text representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7235–7252, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

- Schwarm, S. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530, Ann Arbor, Michigan. Association for Computational Linguistics.
- Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., Staiano, J., Wang, A., and Gallinari, P. (2021). QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *ACL*.
- Sellam, T., Das, D., and Parikh, A. (2020). Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.
- Sennrich, R., Haddow, B., and Birch, A. (2016a). 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. Association for Computational Linguistics.
- Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96.
- Sennrich, R., Haddow, B., and Birch, A. (2016c). 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.
- Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.
- Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2017). Bidirectional attention flow for machine comprehension. In *ICLR*.
- Settles, B. (2012). *Active Learning*, volume 18. Morgan & Claypool Publishers.
- Shapira, O., Gabay, D., Gao, Y., Ronen, H., Pasunuru, R., Bansal, M., Amsterdamer, Y., and Dagan, I. (2019). Crowdsourcing lightweight pyramids for manual summary evaluation. In *NAACL*, pages 682–687.
- Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.
- Smit, P., Virpioja, S., Grönroos, S.-A., and Kurimo, M. (2014). Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

- Smith, L. T. (1999). *Decolonizing Methodologies: Research and Indigenous Peoples*. ERIC.
- Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.
- Song, L., Wang, Z., and Hamza, W. (2017). A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058*.
- Song, L., Wang, Z., Hamza, W., Zhang, Y., and Gildea, D. (2018). Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574.
- Specia, L., Raj, D., and Turchi, M. (2010). Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.
- Spring, J. (2016). Native americans: Deculturalization, schooling, globalization and inequality. In *Deculturalization and the Struggle for Equality*, pages 40–59. Routledge.
- Stahlberg, F. and Byrne, B. (2019). On NMT search errors and model errors: Cat got your tongue? 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 3356–3362, Hong Kong, China. Association for Computational Linguistics.
- Stratos, K., Collins, M., and Hsu, D. (2016). Unsupervised part-of-speech tagging with anchor hidden Markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257.
- Su, Y., Lan, T., Wang, Y., Yogatama, D., Kong, L., and Collier, N. (2022). A contrastive framework for neural text generation. *arXiv preprint arXiv:2202.06417*.
- Subramanian, S., Wang, T., Yuan, X., Zhang, S., Trischler, A., and Bengio, Y. (2018). Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88.
- Sun, L., Hashimoto, K., Yin, W., Asai, A., Li, J., Yu, P., and Xiong, C. (2020). Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*.
- Sun, S., Shapira, O., Dagan, I., and Nenkova, A. (2019). How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sun, X., Liu, J., Lyu, Y., He, W., Ma, Y., and Wang, S. (2018). Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

- Susanti, Y., Tokunaga, T., Nishikawa, H., and Obari, H. (2018). Automatic distractor generation for multiple-choice english vocabulary questions. *Research and practice in technology enhanced learning*, 13(1):1–16.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Tam, D., Mascarenhas, A., Zhang, S., Kwan, S., Bansal, M., and Raffel, C. (2023). Evaluating the factual consistency of large language models through summarization. *Findings of the 61th Annual Meeting of the Association for Computational Linguistics*.
- Tan, H., Yu, L., and Bansal, M. (2019). Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*.
- Tang, D., Duan, N., Qin, T., Yan, Z., and Zhou, M. (2017). Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.
- Tang, D., Duan, N., Yan, Z., Zhang, Z., Sun, Y., Liu, S., Lv, Y., and Zhou, M. (2018). Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1564–1574.
- Tay, Y., Tran, V. Q., Ruder, S., Gupta, J., Chung, H. W., Bahri, D., Qin, Z., Baumgartner, S., Yu, C., and Metzler, D. (2021). Charformer: Fast character transformers via gradient-based subword tokenization.
- Theis, L., van den Oord, A., and Bethge, M. (2016). A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, pages 1–10.
- Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 2214–2218.
- Tratz, S. and Hovy, E. (2008). Bewte: basic elements with transformations for evaluation. In *TAC 2008 Workshop*.

- Tsunoda, T. (2013). *Language endangerment and language revitalization*. De Gruyter Mouton.
- Uchihara, H. (2016). *Tone and accent in Oklahoma Cherokee*, volume 3. Oxford University Press.
- Uibo, H., Pruulmann-Vengerfeldt, J., Rueter, J., and Iva, S. (2015). Oahpa! õpi! opiq! developing free online programs for learning Estonian and Võro. In *Proceedings of the fourth workshop on NLP for computer-assisted language learning*, pages 51–64, Vilnius, Lithuania. LiU Electronic Press.
- Vajjala, S. and Meurers, D. (2012). On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.
- Van Handel, R. (2014). Probability in high dimension. Technical report, PRINCETON UNIV NJ.
- Vasilyev, O., Dharnidharka, V., and Bohannon, J. (2020). Fill in the blanc: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.
- Von Ahn, L. (2008). Human computation. In *2008 IEEE 24th international conference on data engineering*, pages 1–2. IEEE.
- Wan, D. and Bansal, M. (2022). FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.
- Wang, A., Cho, K., and Lewis, M. (2020a). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.
- Wang, D., Liu, P., Zheng, Y., Qiu, X., and Huang, X. (2020b). Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.
- Wang, W., Yang, N., Wei, F., Chang, B., and Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.
- Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2020). Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

- Whalen, D. H., Moss, M., and Baldwin, D. (2016). Healing through language: Positive physical health effects of indigenous language use. *F1000Research*, 5(852):852.
- Wieting, J., Berg-Kirkpatrick, T., Gimpel, K., and Neubig, G. (2019). Beyond BLEU: training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.
- Williams, A., Nangia, N., and Bowman, S. (2018a). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Williams, A., Nangia, N., and Bowman, S. (2018b). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv*.
- Xenouelas, S., Malakasiotis, P., Apidianaki, M., and Androutsopoulos, I. (2019). Sum-qe: a bert-based summary quality estimation model. 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 6007–6013.
- Xie, Y., Sun, F., Deng, Y., Li, Y., and Ding, B. (2021). Factual consistency evaluation for text summarization via counterfactual estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 100–110, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xu, J., Gan, Z., Cheng, Y., and Liu, J. (2020a). Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

- Xu, J., Tan, X., Ren, Y., Qin, T., Li, J., Zhao, S., and Liu, T.-Y. (2020b). Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.
- Xu, J., Zhou, H., Gan, C., Zheng, Z., and Li, L. (2021). Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of ACL 2021*.
- Xu, S., Zhang, X., Wu, Y., and Wei, F. (2022). Sequence level contrastive learning for text summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11556–11565.
- Xu, X., Dušek, O., Li, J., Rieser, V., and Konstas, I. (2020c). Fact-based content weighting for evaluating abstractive summarisation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5071–5081.
- Xu, X. and Ingason, A. K. (2021). Developing Flashcards for learning Icelandic. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 55–61, Online. LiU Electronic Press.
- Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. (2021a). Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021b). 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.
- Yang, Q., Passonneau, R., and De Melo, G. (2016). Peak: Pyramid evaluation via automated knowledge extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Yang, Z., Hu, J., Salakhutdinov, R., and Cohen, W. (2017). Semi-supervised qa with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050.
- Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., and Yan, R. (2019). Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.
- Yuan, X., Wang, T., Gulcehre, C., Sordoni, A., Bachman, P., Zhang, S., Subramanian, S., and Trischler, A. (2017). Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25.
- Zhang, H., Duckworth, D., Ippolito, D., and Neelakantan, A. (2021a). Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

- Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020a). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Zhang, S. and Bansal, M. (2019). Addressing semantic drift in question generation for semi-supervised question answering. 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 2495–2509, Hong Kong, China. Association for Computational Linguistics.
- Zhang, S. and Bansal, M. (2021). Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhang, S., Chaudhary, V., Goyal, N., Cross, J., Wenzek, G., Bansal, M., and Guzman, F. (2022a). How robust is neural machine translation to language imbalance in multilingual tokenizer training? In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 97–116, Orlando, USA. Association for Machine Translation in the Americas.
- Zhang, S., Frey, B., and Bansal, M. (2020b). ChrEn: Cherokee-English machine translation for endangered language revitalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 577–595, Online. Association for Computational Linguistics.
- Zhang, S., Frey, B., and Bansal, M. (2021b). ChrEnTranslate: Cherokee-English machine translation demo with quality estimation and corrective feedback. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 272–279, Online. Association for Computational Linguistics.
- Zhang, S., Frey, B., and Bansal, M. (2022b). How can NLP help revitalize endangered languages? a case study and roadmap for the Cherokee language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1529–1541, Dublin, Ireland. Association for Computational Linguistics.
- Zhang, S., Mahmut, G., Wang, D., and Hamdulla, A. (2017). Memory-augmented chinese-uyghur neural machine translation. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1092–1096. IEEE.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022c). Opt: Open pre-trained transformer language models.

- Zhang, S., Wan, D., and Bansal, M. (2023a). Extractive is not faithful: An investigation of broad unfaithfulness problems in extractive summarization. *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.
- Zhang, S., Wu, S., Irsoy, O., Lu, S., Bansal, M., Dredze, M., and Rosenberg, D. (2023b). Mixce: Training autoregressive language models by mixing forward and reverse cross-entropies. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020c). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., and Eger, S. (2019). Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578.
- Zhao, Y., Ni, X., Ding, Y., and Ke, Q. (2018). 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.
- Zheng, H. and Lapata, M. (2019). Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.
- Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.
- Zhong, M., Liu, P., Wang, D., Qiu, X., and Huang, X. (2019). Searching for effective neural extractive summarization: What works and what’s next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.
- Zhou, D., Guo, L., and He, Y. (2018a). Neural storyline extraction model for storyline generation from news articles. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1727–1736, New Orleans, Louisiana. Association for Computational Linguistics.
- Zhou, L., Kwon, N., and Hovy, E. (2007). A semi-automatic evaluation scheme: Automated nuggetization for manual annotation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 217–220.

- Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., and Zhao, T. (2018b). Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.
- Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., and Zhou, M. (2017). Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.
- Zhou, Z.-H. and Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.
- Zhu, J., Xia, Y., Wu, L., He, D., Qin, T., Zhou, W., Li, H., and Liu, T. (2020). Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.
- Ács, J. (2019). Exploring bert’s vocabulary. In *Judit Ács’s blog*, Online.