ABSTRACT

Title of Dissertation: Stronger Inductive Biases for Sample-Efficient and Controllable Neural Machine Translation

Weijia Xu
Doctor of Philosophy, 2023

Dissertation Directed by: Professor Marine Carpuat
Department of Computer Science

As one of the oldest applications of natural language processing, machine translation (MT) has a growing impact on human lives both as an end application and as a key component of cross-lingual information processing such as cross-lingual information retrieval and dialogue generation. Although neural machine translation (NMT) models achieve impressive performance on some language pairs, they are trained on large amounts of human translations. In addition, they are notorious for generating fluent outputs that do not faithfully reflect the meaning of the source sentence, and they make it difficult for users to control the outputs. To address these issues, this thesis contributes techniques to build more sample-efficient and controllable NMT models by incorporating stronger inductive biases that help correct undesirable biases, integrate prior knowledge, and introduce flexible ways to control the outputs in NMT.

In our first line of research, we show that current NMT models are susceptible to undesirable biases that hinder sample-efficient training and lead to unfaithful translations. We further provide evidence that we can mitigate these undesirable biases by integrating stronger inductive biases

through training algorithms. We start by introducing a new training objective to address the exposure bias problem – a common problem in sequence generation models that typically causes accumulated errors along the generated sequence at inference time, especially when the training data is limited. Next, we turn to a well-known but less studied problem in MT – the hallucination problem – translation outputs that are unrelated to the source text. To find spurious biases that cause hallucination errors, we first identify model symptoms that are indicative of hallucinations at inference time. And then, we show how these symptoms connect to the spurious biases at training time, where the model learns to predict the ground-truth translation while ignoring a large part of the source sentence. These findings provide a future path toward mitigating hallucinations by addressing these spurious biases.

In our second line of research, we study how to integrate stronger inductive biases in NMT for effective integration of the language priors estimated from unsupervised data. We introduce a novel semi-supervised learning objective with a theoretical guarantee on its global optimum and show that it can be effectively approximated and leads to improved performance in practice.

Finally, we study inductive biases in the form of NMT model architectures to allow end users to control the model outputs more easily. Controlling the outputs of standard NMT models is difficult with high computational cost at training or inference time. We develop an edit-based NMT model with novel edit operations that can incorporate users' lexical constraints with low computational cost at both training and inference time. To allow users to provide lexical constraints in more flexible morphological forms, we further introduce a modular framework for inflecting and integrating lexical constraints in NMT.

Stronger Inductive Biases for Sample-Efficient and Controllable Neural Machine Translation

by

Weijia Xu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Marine Carpuat, Chair/Advisor
Professor Hal Daumé III
Professor Soheil Feizi
Professor He He
Professor Doug Oard (Dean's Representative)

© Copyright by
Weijia Xu
2023

Acknowledgments

I am very fortunate to have got the opportunity to pursue my PhD at the University of Maryland (UMD). I really enjoyed my time at UMD, and I am deeply grateful to a lot of people who have helped and supported me throughout the journey. Their support helped me craft this dissertation and grow as a researcher.

First, I would like to thank my wonderful advisor, Marine Carpuat, for her guidance and support throughout the years. She taught me many things, all of which have contributed to making me a better researcher. She guided me through my very first academic paper and taught me how to motivate the work in light of the literature and design experiments to answer the right research questions. She also encouraged me to constantly improve my writing and presentation skills. I am also very grateful for the immense flexibility she gave me to choose the research problems that I wanted to work on, while providing timely feedback and sharp insights along the way. Apart from research, I am also thankful to Marine for her career advice and for encouraging me to contribute to diversity and inclusion in the research community.

I would also like to thank the members of my dissertation committee – Hal Daumé III, Soheil Feizi, He He and Doug Oard – for their valuable feedback and insightful comments to make this dissertation deeper and more solid. A special thank you to Doug Oard for his great support and constructive feedback on our work in the MATERIAL program, and to He He for agreeing to be an external member on my committee. I am also very fortunate to have been a

part of the amazing Computational Linguistics and Information Processing (CLIP) lab at UMD. I learned a lot from the weekly CLIP talks, reading groups and paper clinics. I thank all the faculty members of the CLIP lab for organizing all these events and fostering such an inclusive environment for the students at the CLIP lab. I would also like to thank all the amazing students at the CLIP lab, without whom my PhD life would not have been as enjoyable. Among them I would like to thank are: Xing Niu, Sweta Agrawal, Eleftheria Briakou and Marianna Martindale for being such amazing collaborators; Yogarshi Vyas, Yifan Yang, HyoJung Han, Calvin Bao, Aquia Richburg, Yow-Ting Shiue and Abigail Oppong for their timely feedback on my paper drafts and presentations; Pedro Rodriguez and Michelle Yuan for helping me through the job search; Elijah Rippeth and Shramay Palta for organizing the CLIP social events; and many former and current CLIP members including Kianté Brantley, Pranav Goel, Peter Rankel, Alexander Hoyle, Matthew Shu, Fenfei Guo, Ahmed Elgohary, Denis Peskov, Suraj Nair, Amr Sharaf, Khanh Nguyen, Shi Feng, Weiwei Yang, Joe Barrow, Petra Galuscakova, Chen Zhao, Trista Cao, Chenglei Si, Yoo Yeon Sung, Sathvik Nair, Sandra Sandoval, Neha Srikanth, Navita Goyal and Maharshi Gor. Finally, I would also like to thank Sharron McElroy, Vivian Lu, Jodie Gray and Tom Hurst at UMD for making all the administrative tasks easy.

I am also fortunate to have got the opportunities to work with fantastic industry mentors through internships. At Amazon, I am grateful to Saab Mansour and Batool Haider for giving me the freedom to explore various research ideas and helping me improve the design of the experiments to better answer our research questions. At Microsoft Research, I had many insightful discussions with Yuwei Yin, Shuming Ma, Dongdong Zhang and Haoyang Huang. I am also thankful to Jiatao Gu, Wei-Ning Hsu and Marjan Ghazvininejad at Meta AI Research, who had been a great source of inspiration for me to frame research projects in a way that makes

broader impacts.

Close friends and family have been a precious source of love and support through the ups and downs of this journey. To Qian Xu, Jing Ye, Liyun Tu and Ziyi Yuan, thank you for being such amazing roommates and for making me feel at home away from my home. To Yuxin Xiong, Danxuan Chen, Rufei Li, Yu Wang, Fenfei Guo, Xing Niu, Sweta Agrawal and Eleftheria Briakou, thank you for all your love and support. My PhD life would not have been so enjoyable without you guys. I would also like to thank my badminton and soccer teammates – Siyuan, Jiayang Sun, Eddie Li, Jerry Yin, Ziyang Shen, Shiyuan Xiang, Xuan Su, Bosheng Li and Shenghao Li – for all the fun times and incredible memories. Last but not the least, I thank my parents for their love and support throughout the years. They are always there for me, although I was physically away from them most of the time.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Machine translation (MT) is the task of translating text from one language to another using computational methods. It is one of the oldest applications of natural language processing and has a growing impact on human lives. Through online MT services such as Google Translate (Wu et al., 2016), it allows users to communicate across language barriers and acquire knowledge conveyed in different languages. MT is used not only as an end application, for instance to translate an English web page into Chinese, but also as a component of other language technology applications to find documents relevant to a query even when they are written in different languages (Oard, 1998; Oard et al., 2019) and to summarize documents written in multiple languages (Nguyen and Daumé III, 2019; Ouyang et al., 2019). As the current defacto standard in MT, neural machine translation (NMT) models the translation process using a single neural network trained on pairs of source and ground-truth translation sentences. NMT has achieved outstanding performance and outperformed other types of MT models on many resource-rich language pairs such as English-French and English-Chinese (Hassan et al., 2018; Wu et al., 2016). However, to make MT accessible and useful for all, we need to build MT models with reasonable translation quality not only for a small number of languages with rich resources, but for languages with limited data and human resources as well. In addition, MT models should be able to generate translations controlled by the source sentence and users' preferences. Unfortunately, current NMT

models still struggle to meet these criterion. First, NMT models are typically data hungry – they rely on large amounts of supervised training data to achieve reasonable performance. This is unsatisfactory since most world languages are low-resource languages with very limited data and human resources. Second, the end-to-end nature of NMT has made it difficult to control, and NMT models are notorious for generating fluent but unfaithful translations. This type of error can lead to significant harm. For example, the mistaken translation of a social media post that meant "Good morning" into "Attack them" by an MT service has led to the wrongful arrest of the post's author (Berger, 2017).

A fundamental cause of these problems in NMT is that NMT models have weak inductive biases. Inductive biases are priori assumptions about which solutions are preferred (Mitchell, 1980). Before the advent of NMT, rule-based MT approaches incorporated inductive biases in the form of human-crafted linguistic rules that guide the translation process (Forcada et al., 2011; Mayor et al., 2011). Later on, statistical MT models were developed in which inductive biases are integrated in the design of the feature functions and model components (e.g. phrasal translation probabilities, alignment probabilities, target-side language models, etc.) used to compute the conditional probability of a translation given a source sentence (Berger et al., 1994; Brown et al., 1990; Koehn et al., 2003). In an attempt to train a single model in an end-to-end fashion, however, NMT models the translation probabilities using a single neural network which typically has weak inductive biases: they view translation as a probabilistic transformation from a source sequence to a target sequence with little assumptions about the intermediate steps of the translation process (Bahdanau et al., 2015; Sutskever et al., 2014). This facilitates effective learning on huge amounts of supervised data, but also leads to overly expressive models that can learn many different ways to map the source sentences into the target language, some of which

are more desirable than the others. As a result, NMT models are extremely data hungry – they need to be trained on large amounts of supervised data to learn a plausible mapping from the source to the target language. It also makes it difficult for users to control how an NMT model would translate a source sentence among all possible ways that it may learn from data.

Prior work has shown that combining the power of deep neural networks with prior knowledge as inductive biases can mitigate spurious correlations in NMT models and improve their translation quality. Such inductive biases can be semantic or syntactic knowledge extracted from external resources. For instance, Campolungo et al. (2022) show that integrating word sense annotations extracted from a multilingual knowledge base (Navigli and Ponzetto, 2012) into NMT reduces the spurious biases in word sense disambiguation. Other works have also incorporated syntactic information about the source and target languages as a type of inductive bias in NMT, which helps the models handle the syntactic divergences between source and target and leads to improved translation quality especially when supervised data is limited (Currey and Heafield, 2019; Zhang et al., 2019; Zhou et al., 2019). However, for many low-resource languages, collecting such linguistic resources can be very expensive or even infeasible. In this dissertation, we focus instead on integrating inductive biases about the translation process itself without requiring any manually annotated linguistic data.

1.1 Research Questions

In this dissertation, we ask the question: how can we incorporate stronger inductive biases for more sample-efficient and controllable NMT? To make NMT more sample-efficient, we need to properly guide the models in the large learning space and steer them away from the undesirable

biases that may prohibit them from generalizing beyond the training samples (e.g. exposure bias and spurious biases). To this end, we ask if we can introduce stronger inductive biases in the form of supervised learning algorithms to mitigate these undesirable biases. Furthermore, as another way to approach sample-efficient NMT, we research semi-supervised learning algorithms that help models exploit unsupervised data, which is typically more abundant and less expensive to collect than supervised data. Next, we explore ways to strengthen the inductive biases for more controllable NMT. For this, we ask if stronger inductive biases can be introduced through more flexible model architectures to incorporate users' lexical preferences in the outputs more easily.

1.2 Roadmap

This dissertation is organized into eight chapters. We first discuss relevant background on machine translation, interpretation methods of natural language processing models, and existing methods to strengthen the inductive biases in text generation models in Chapter 2. Next, we describe the details of our work in Chapter 3–7. Finally, we conclude in Chapter 8 by summarizing our main contributions, discussing the limitations, and suggesting future directions based on this work.

1.2.1 Stronger Inductive Biases for Mitigating Undesirable Biases

We first study how to integrate stronger inductive biases through supervised learning algorithms to combat the undesirable biases in NMT models. We focus on the undesired biases that hamper the models' ability to generalize beyond the training samples and thus lead to low sample efficiency and unfaithful translations. In Chapter 3 and 4, we identify the undesired

biases that cause these issues and introduce methods to mitigate some of these undesirable biases by strengthening the models' inductive biases.

In Chapter 3, we study the exposure bias problem – a well-known undesirable bias that hamper the models' ability to generalize from training to inference (Bahdanau et al., 2015; Sutskever et al., 2014). The problem is caused by a gap between training and inference: during training the model is only exposed to the ground-truth prefixes, but has to rely on its predicted prefixes at inference time. This typically causes accumulated errors along the generated sequence at inference time, especially when the training corpus is small. We propose to mitigate exposure bias by integrating stronger inductive biases through a novel learning algorithm that bridges the gap between training and inference, while allowing the model to generate sentences with different word orders than the ground-truth sequence. Experiments on three translation tasks show that our learning algorithm alleviates the exposure bias problem and significantly improves translation quality over existing methods. This work was previously published in Xu et al. (2019).

In Chapter 4, we change our focus to the spurious bias problems that lead to hallucinations – an egregious type of error in NMT where the translation outputs are unrelated to the source text. Hallucinations are potentially harmful but less studied in the field – it remains unclear in what conditions they arise and how to mitigate them. We argue that a potential cause of the hallucination problem is the spurious biases – prediction rules that work well for a subset of the training samples but do not generalize to other samples. To trace hallucinations back to the spurious biases, we first identify internal model symptoms of hallucinations at inference time. Next, we show that these symptoms are reliable indicators of various types of hallucinations, either those under source-side perturbations or the ones on natural inputs. Finally, we trace these symptoms to the spurious biases at training time, where the model learns to predict the ground-

truth translation while ignoring a large part of the source sentence. These findings pave the road for future studies to reduce hallucinations by introducing stronger inductive biases to combat these spurious biases.

1.2.2 Stronger Inductive Biases for Integrating Language Priors

In low-resource scenarios where the amount of supervised data is limited, learning from unsupervised monolingual data can reduce the models' reliance on supervised data and thus improve sample-efficiency. In Chapter 5, we investigate the inductive biases from semi-supervised learning algorithms for effective integration of the language priors estimated from monolingual data. Existing approaches to integrating prior language distributions into NMT (e.g. Back-Translation (Sennrich et al., 2016b), Iterative Back-Translation (Zhang et al., 2018), and Dual Learning (He et al., 2016)) are mostly based on heuristics. Little is known about why they work or how they compare. To answer these questions, we introduce a theoretical framework, namely Dual Reconstruction Objective (DRO), that unifies these approaches and explains why they work. Specifically, we prove that DRO shares the same global optimum as the ideal but intractable objective that maximizes the marginal likelihood of observed sequences from monolingual data. Based on this framework, we provide theoretical and empirical comparisons between existing approaches, which shed new lights on more effective integration of prior language distributions in NMT. This work was previously published in Xu et al. (2020).

1.2.3 Stronger Inductive Biases for Controllable NMT

In the final part of the research, we study inductive biases in the form of model architectures toward controllable NMT. Current NMT models typically share the same inductive bias that a translation sequence should be predicted token by token from left to right, which makes it difficult for users to control the models' outputs based on their knowledge and preferences. To address this problem, we introduce new model architectures to better incorporate the users' lexical or phrasal preferences in models' outputs in Chapter 6 and 7.

In Chapter 6, we introduce a more controllable model architecture to allow users to inject knowledge and communicate their intentions to the model. Standard autoregressive NMT models generate a sentence token-by-token from left to right, which makes it difficult for users to control the generation process. By contrast, edit-based models that generate sequences through explicit edit operations provide a more flexible paradigm for controllable generation. We develop EDITOR, an edit-based transformer model with a novel reposition operation that can seamlessly incorporate users' lexical preferences with low computational cost at training and inference time. Our evaluation on three machine translation benchmarks shows that EDITOR generates texts in a more controllable way by incorporating users' lexical preferences more effectively than the state-of-the-art edit-based NMT model, and the generation process is more efficient compared to autoregressive models. This work previously appeared in Xu and Carpuat (2021a).

In Chapter 7, we further extend this approach to incorporate users' preferences or constraints in more flexible morphological forms by introducing a modular framework for inflecting and incorporating constraint terms in NMT. Previously, we were focusing on incorporating constraint terms in the exact forms as given. However, in real-world application

scenarios, it is difficult for users to infer the exact morphological forms of the constraints without looking at the final translation. Thus, we propose a modular framework that takes constraint terms in dictionary forms, inflects them based on the sentence-level context, and incorporates them in the final translation. Based on this framework, we show the complementary capabilities of neural and rule-based models, which are typically difficult to integrate together. We achieve the best results by using a rule-based module to derive the inflection forms of constraints based on known linguistic rules and powerful neural models to integrate the inflected constraints into the final translations. This work was previously published in Xu and Carpuat (2021c).

1.3 Contributions

This dissertation makes the following contributions:

- We address the exposure bias in NMT by introducing a supervised learning algorithm that trains NMT models to predict the next token based on their previous predictions, while allowing for word order differences between the models' outputs and the ground-truth translations (Chapter 3).
- We discover the spurious biases that are related to the hallucination phenomenon in NMT by first identifying the model symptoms that are reliable indicators of hallucinations under various conditions and then connecting the symptoms to the spurious biases learned at training time (Chapter 4).
- We contribute a better understanding of the inductive biases embedded in existing semi-supervised learning algorithms by introducing a theoretical framework with a guarantee on its global optimum to unify and compare existing algorithms. The theory justifies

the use of a simple learning algorithm instead of the more complex one, which is supported by a systematic empirical comparison of these learning algorithms in three data settings (Chapter 5).

- We introduce EDITOR, a novel edit-based transformer model based on insertion and reposition operations with strong inductive biases to support controllable MT with lexical constraints. EDITOR exploits lexical constraints more effectively than the state-of-the-art edit-based model and more efficiently than the traditional autoregressive model (Chapter 6).
- We further design a modular framework to allow users to control the NMT outputs more flexibly by providing lemma-form constraints without inflections. This framework is applicable to diverse types of NMT architectures and inflection modules, including neural-based and rule-based ones, and facilitates fast adaptation to unseen constraint terms (Chapter 7).
- We construct new test suites to evaluate models' ability to inflect and incorporate lemma-form constraints on English-German and English-Lithuanian translation tasks (Chapter 7).
- We release the implementations of the aforementioned models and algorithms, along with the new datasets at `https://github.com/weijia-xu`.

Chapter 2: Background

This work draws on many different areas in machine translation and natural language processing in general. We first discuss the fundamental models and main challenges in machine translation. We then discuss the literature related to integrating inductive biases in text generation models via training algorithms and generation schemes. Finally, we discuss the literature relevant to our work on interpretation of natural language processing models.

2.1 Machine Translation

Machine Translation (MT) is the task of translating text from one natural language into another. Modern MT systems are typically data driven – they rely on models trained on bilingual corpora consisting of texts in the source language paired with their translations in the target language. More formally, an MT model computes the probability of a target translation $\boldsymbol{y}$ given a source sentence $\boldsymbol{x}$ as $P(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the model parameters. The inference process aims to find the most probable translation:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) \tag{2.1}$$

In this section, we will discuss the fundamental models in MT, as well as the current challenges.

2.1.1 Fundamental Models

The conditional probability $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ in MT can be modeled using various types of models. In this section, we will introduce the two dominant models – Statistical Machine Translation (SMT) (Berger et al., 1994; Brown et al., 1990; Koehn et al., 2007, 2003) (among which Phrase-Based Machine Translation (PBMT) is the dominant approach) and Neural Machine Translation (NMT) (Bahdanau et al., 2015; Hassan et al., 2018; Sutskever et al., 2014). We will also introduce Language Models (LMs), which serve as an important component in PBMT and some NMT models.

2.1.1.1 Phrase-Based Machine Translation

Phrase-Based Machine Translation models the probability $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ of a target translation $\boldsymbol{y}$ given a source sentence $\boldsymbol{x}$ at the granularity of contiguous sequences of words, namely phrases, using a standard log linear model as follows:

$$P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \frac{\exp\sum_i \lambda_i h_i(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'}\exp\sum_i \lambda_i h_i(\boldsymbol{x},\boldsymbol{y}')} \tag{2.2}$$

where h_i are feature functions and λ_i are feature weights. Each feature function is learned independently on the training set, while the feature weights are optimized on the development set. The core features include phrase translation probabilities aggregated from word-level alignments (i.e. the mapping between source and target words automatically learned from the bilingual training corpora), target language model (i.e. probability distribution over the target sentences learned from monolingual corpora to improve the fluency of the output),

relative distortion probability distribution (i.e. a learned function over the position difference between aligned phrase pairs), word penalty that calibrates the output length, etc. Feature weights are estimated by maximizing a translation quality metric (e.g. BLEU (Papineni et al., 2002)) on the development set using optimization algorithms such as Minimum Error Rate Training (MERT) (Och, 2003), Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003) and batch MIRA (Cherry and Foster, 2012).

To generate a translation, ideally we search for the target sentence with the maximum conditional probability given the PBMT model. In practice, it is intractable to consider all collections of phrases. Beam search has proven to be a good approximation that balances efficiency with translation quality (Koehn, 2004; Koehn et al., 2003).

2.1.1.2 Neural Machine Translation

Different from PBMT that rely on multiple modules learned independently on the training set, Neural Machine Translation (NMT) models the conditional probability $P(\boldsymbol{y}\,|\,\boldsymbol{x};\boldsymbol{\theta})$ using a single neural network with parameters $\boldsymbol{\theta}$ that are trained end-to-end. The current defacto standard NMT models generate the target sequence token by token from left to right, i.e. they factorize the probability of the target sequence as:

$$P(\boldsymbol{y}\,|\,\boldsymbol{x}) = \prod_{t=1}^{T} p(y_t\,|\,\boldsymbol{y}_{<t},\boldsymbol{x};\theta) \tag{2.3}$$

where the probability of the target token y_t at step t is conditioned on the target prefix $\boldsymbol{y}_{<t}$ and the source sentence $\boldsymbol{x}$. This probability is typically parameterized by an encoder-decoder network (Cho et al., 2014b; Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014), where

an encoder first encodes the source sentence into a sequence of hidden representations $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_n)$:

$$\boldsymbol{C} = \text{Encoder}(\boldsymbol{x}) \tag{2.4}$$

where n is the length of the source sequence.

Next, a decoder produces a hidden representation $\boldsymbol{h}_t$ at each step t given the target prefix $\boldsymbol{y}_{<t}$ and the source representations $\boldsymbol{C}$:

$$\boldsymbol{h}_t = \text{Decoder}(\boldsymbol{y}_{<t}, \boldsymbol{C}) \tag{2.5}$$

At each time step t, the hidden representation $\boldsymbol{h}_t$ is fed to a linear projection layer $\boldsymbol{s}_t = \boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}$ to obtain a vector of scores $\boldsymbol{s}_t$ over all possible words in the vocabulary $\mathcal{V}$. Scores are then turned into a conditional probability distribution:

$$p(\cdot \,|\, \boldsymbol{y}_{<t}, \boldsymbol{x}; \theta) = \text{softmax}(\boldsymbol{s}_t) \tag{2.6}$$

The encoder and decoder can be parameterized via different neural model architectures, such as Recurrent Neural Networks (RNNs) (Cho et al., 2014b; Sutskever et al., 2014), Convolutional Neural Networks (CNNs) (Gehring et al., 2017), and self-attentional Transformer (Vaswani et al., 2017). RNNs are natural to model variable-length sequences but are faced with challenges when modeling long-distance dependencies. In addition, the hidden representations are computed recurrently, which limits its parallelization. Instead, CNNs use convolutional layers to combine a sequence of representations in a limited window into a single representation. More recently, Vaswani et al. (2017) proposed the Transformer architecture that

models the dependencies between all tokens in the source or target sequence through multi-head self-attention, where a single attention head produces a context matrix given a query matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times d}$, a key matrix $\boldsymbol{K} \in \mathbb{R}^{n \times d}$, and a value matrix $\boldsymbol{V} \in \mathbb{R}^{n \times d}$ (d is the dimension of hidden representations):

$$\boldsymbol{C}_i = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{W}_i^Q(\boldsymbol{K}\boldsymbol{W}_i^K)^\top}{\sqrt{d_i}}\right)\boldsymbol{V}\boldsymbol{W}_i^V \tag{2.7}$$

where weight matrices $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$, and $\boldsymbol{W}_i^V$ belong to $\mathbb{R}^{d \times d_i}$, $d_i = d/h$ is the dimension of each attention head, and h is the number of attention heads. Finally, the final context matrix is computed by concatenating the attention heads followed by a linear projection:

$$\text{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = [\boldsymbol{C}_1,\boldsymbol{C}_2,...,\boldsymbol{C}_h]\boldsymbol{W}^O \tag{2.8}$$

Given a sequence of hidden representations $\boldsymbol{H} = [\boldsymbol{h}_1^{\text{enc}},...,\boldsymbol{h}_n^{\text{enc}}]$ from the previous layer, the encoder first computes self-attention via $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{H}$, followed by a point-wise feed-forward network, a post-processing stack of dropout, layer normalization (Ba et al., 2016) and residual connection. The decoder follows the same computation steps except for an additional source attention layer, which uses the decoder representations $\boldsymbol{H}$ as query, and the source representations $\boldsymbol{C}$ as key and value to compute the multi-head attention.

Supervised NMT models are typically trained via the traditional maximum likelihood objective that maximizes the log-likelihood of the training data $\mathscr{D} \equiv \{(\boldsymbol{x}^{(n)},\boldsymbol{y}^{(n)})\}_{n=1}^N$ consisting

of N pairs of source and target sentences:

$$\mathscr{J}_{ML}(\theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(y_t^{(n)} \,|\, \boldsymbol{y}_{<t}^{(n)}, \boldsymbol{x}^{(n)}; \theta) \tag{2.9}$$

At test time, to find the sequence of target tokens with the maximum probability, NMT also uses beam search as an approximation similarly to PBMT.

2.1.1.3 Language Model

To improve the fluency of the output text, PBMT include a language model on the target language as a feature function. A language model estimates the probabilities of token sequences $P(x_1 ... x_m)$, which can be decomposed using the chain rule:

$$P(x_1 ... x_m) = \prod_{k=1}^{m} P(x_k \,|\, x_{1...k-1}) \tag{2.10}$$

The probabilities can be modeled in various ways. For example, PBMT typically uses N-gram LMs, which assume that the probability of a token can be predicted based on the previous $N-1$ tokens:

$$P(x_k \,|\, x_{1...k-1}) \approx P(x_k \,|\, x_{k-N+1...k-1}) \tag{2.11}$$

The probabilities are then estimated to maximize the likelihood of token sequences in a text corpus:

$$P(x_k \,|\, x_{k-N+1...k-1}) = \frac{C(x_{k-N+1...k})}{C(x_{k-N+1...k-1})} \tag{2.12}$$

More recently, neural models have achieved new state-of-the-art on language modeling.

These models compute the probability of a token given all previous tokens using RNNs (Mikolov et al., 2010b) or Transformer models (Radford et al., 2018). In Section 2.1.3, we will show that neural LMs can be integrated in NMT models to improve the quality of translation outputs.

2.1.2 Automatic Evaluation for Machine Translation

Human evaluation of MT is expensive. Prior work has proposed automatic evaluation metrics for MT given one or multiple references for each source sentence, among which the most widely-used evaluation metric is BLEU (Papineni et al., 2002). BLEU is computed based on the n-gram precision p_n of translations compared against the references and brevity penalty BP:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{otherwise} \end{cases} \tag{2.13}$$

where c and r are the lengths of the translation output and reference, respectively. The resulting BLEU score is then:

$$\text{BLEU} = BP \cdot \exp(\sum_n w_n \log p_n) \tag{2.14}$$

where n represents the orders of n-gram considered for p_n and w_n represents the weights assigned for the n-gram precisions. In practice, the weights are typically uniform.

One drawback of BLEU is the lack of explicit consideration for reordering, which is problematic when evaluating translations between two distant languages with completely different word order. To overcome this problem, Isozaki et al. (2010) introduce the RIBES score, which is computed based on the unigram precision and word order metrics which compare the word ranks in the reference with those in the hypothesis.

MT evaluation is an active area of research with many new metrics proposed (Rei et al., 2020a; Sellam et al., 2020; Zhang* et al., 2020), yet BLEU remains the most widely used metric despite its well-documented flaws (Kocmi et al., 2021).

2.1.3 Semi-Supervised Neural Machine Translation

Although NMT has achieved impressive translation quality on many high-resource language pairs (Hassan et al., 2018; Xu and Carpuat, 2018), its translation quality suffers when the amount of bilingual parallel data is limited (Koehn and Knowles, 2017). This is problematic because most of the over 7000 world languages are low-resource languages with very limited data resources. For example, on Tagalog-English, there are only 70K sentence pairs in WMT training data (Wenzek et al., 2021), while high-resource language pairs such as Chinese-English have over 20M sentence pairs in WMT. Although Sennrich and Zhang (2019) show that NMT with carefully-tuned hyper-parameters can still outperform PBMT in low-resource settings, an advantage of PBMT is that its translation quality can be boosted by incorporating large amounts of monolingual data in its target language model, which is especially helpful in low-resource settings. For NMT, however, incorporating monolingual data effectively in the model is challenging, since it models the conditional probability $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ using a single neural network, and there is no separate language models that can be direct trained on the monolingual data. To address the problem, semi-supervised techniques have been proposed to better leverage large monolingual corpora for NMT. One line of semi-supervised work trains a target LM using monolingual data to improve the fluency of the generated translations. A target LM can be integrated into NMT through LM fusion, which either uses the LM to score candidate tokens

generated by the NMT during training (Stahlberg et al., 2018) or inference (Gulcehre et al., 2015; Skorokhodov et al., 2018), or concatenates the hidden representations from the LM and NMT decoders prior to the output projection layer(Gulcehre et al., 2015). Other approaches use LMs to initialize the embeddings (Abdou et al., 2017), encoder and decoder (Ramachandran et al., 2017) of an NMT model. More recently, initializing NMT parameters using pre-trained multilingual LMs such as multilingual BERT (Devlin et al., 2019) and mBART (Liu et al., 2020a) has led to promising performance, especially on translation directions with little parallel data (Liu et al., 2020a; Zhu et al., 2020).

Another line of work trains NMT models using monolingual and parallel data jointly. A widely used approach is Back-Translation (Sennrich et al., 2016b), which trains the source-to-target translation model $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ by maximizing the conditional log-likelihood of target language sentences $\boldsymbol{y}$ from the monolingual corpora given pseudo source sentences $\tilde{\boldsymbol{x}}$ inferred by a pre-trained target-to-source translation model $P(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\phi})$ given $\boldsymbol{y}$. Iterative Back-Translation optimizes the dual translation models $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ and $P(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\phi})$ via back-translation in turn (Cotterell and Kreutzer, 2018; Hoang et al., 2018; Niu et al., 2018; Zhang et al., 2018). Dual Learning takes the view of cooperative game theory where dual models collaborate with each other to learn to reconstruct the observed source and target monolingual sentences (He et al., 2016). Concretely, Dual Learning optimizes $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ and $P(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\phi})$ jointly by reconstructing the original target sentence $\boldsymbol{y}$ using $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$ given the source $\tilde{\boldsymbol{x}}$ inferred by $P(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\phi})$, and vice versa. The reconstruction loss is augmented with a language model loss and used to update both reconstruction and inference models via policy gradient (Williams, 1992).

Despite their empirical success, there is a lack of theoretical understanding and comparison of these approaches. Cotterell and Kreutzer (2018) interpret Back-Translation as a variational

approximation where the pseudo source $\tilde{\boldsymbol{x}}$ can be viewed as a latent variable and the target-to-source model $P(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\phi})$ is an inference network that approximates the posterior distribution $P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$. Furthermore, they explain Iterative Back-Translation as a way to better approximate the true posterior distribution with the target-to-source model. However, it is unclear how their heuristic objective relates to the ideal objective of maximizing the model's marginal likelihood of the target language monolingual data. More recently, He et al. (2020) connect back-translation and the language model loss in Dual Learning to the variational lower-bound (ELBO) of the marginal likelihood objective. We will introduce a more direct connection through the Dual Reconstruction Objective in Chapter 5.

2.1.4 Lexically Constrained Machine Translation

The end-to-end modeling of NMT (Bahdanau et al., 2015; Vaswani et al., 2017) makes it difficult for users to specify preferences that could be incorporated more easily in statistical MT models (in phrase translation probabilities) (Koehn et al., 2003). For instance, a user may have some preferred words or phrases to appear in the translation, which can be specified as lexical constraints. In cases where the MT model is tested on inputs that contains many domain-specific terminologies (e.g. medical terms) that are rarely seen in the training data, it is helpful to incorporate terminology translations provided by human experts or dictionaries as lexical constraints in MT (Hokamp and Liu, 2017). For example, when an MT model translates the English sentence "The routine use of abdominal drainage to reduce postoperative complications after appendectomy" into German, it may mistranslate the terminology "appendectomy". But this error can be avoided by providing the word "Appendektomie" (translation of "appendectomy")

as a constraint during translation.

Lexical constraints have been incorporated in prior work through two different ways. The first one is constrained training where NMT models are trained on parallel samples with target constraint phrases annotated on the source side by replacing the source phrase with its translation constraint (Song et al., 2019) or inserting the translation constraint next to the source phrase (Dinu et al., 2019). To construct the training data, they extract terminology translations from large term bases and match it with the reference translation. Another approach to incorporating lexical constraints is constrained decoding where beam search is modified to keep track of the number of constraints generated in each beam and to ensure that hypotheses have met all the constraints before they are considered to be completed (Hokamp and Liu, 2017; Post and Vilar, 2018). These mechanisms can incorporate domain-specific knowledge in NMT and reduce translation errors when tested on a new domain (Hokamp and Liu, 2017). However, they suffer from several issues: constrained training requires building dedicated models for constrained generation, while constrained decoding adds significant computational overhead (about 3x slower than unconstrained decoding (Post and Vilar, 2018)) and treats all constraints as hard constraints which may hurt fluency. In other tasks, various constraint types have been introduced by designing complex architectures tailored to specific content or style constraints (Abu Sheikha and Inkpen, 2011; Mei et al., 2016), or via segment-level "side-constraints" (Agrawal and Carpuat, 2019; Ficler and Goldberg, 2017; Sennrich et al., 2016a), which condition generation on users' stylistic preferences, but do not offer fine-grained control over their realization in the output sequence. In Chapter 6, we will introduce an edit-based model that allows for fine-grained control over the outputs through lexical preferences or constraints without increasing the computational cost at training or decoding time.

2.1.5 Hallucinations in Machine Translation

The aforementioned techniques have greatly improved the translation quality of NMT systems. However, NMT still suffers from well-known pathologies such as under-translation (i.e. missing information from the source) and hallucinations, which are pathological translations that are unfaithful to the source (Filippova, 2020; Xiao and Wang, 2021a; Zhou et al., 2021). This include outputs that are completely unrelated to the source (i.e. *detached hallucinations* (Raunak et al., 2021)) and those containing segments that are not supported by the source (which we call *partial hallucinations*). Both types of hallucinations can have critical impact in the real world. For example, given the Chinese source sentence "只有在采取更严格的控制措施时才能发现这种情况" which means "This can only be detected if the controls undertaken are more rigorous", an MT model generates a detached hallucination "Blood alone moves the wheel of history, I say to you and you will understand, it is a privilege to fight." Partial hallucinations can also cause severe problems. For instance, an MT system mistranslates the source "Hold the kidney medicine until you have a chance to talk to your kidney doctor" into " 保留肾药，直到您有机会与您的肾病医生交谈" (Kurtzman, 2019), which means "Keep taking the kidney medicine until you have a chance to talk to your kidney doctor" and may misguide the patient.

To better understand how hallucinations are generated, Raunak et al. (2021) study two types of detached hallucinations: hallucinations triggered by adding an additional token into a natural source sentence and natural hallucinations given unperturbed input sentences. They explain hallucinations from the perspective of training data. For the first type of hallucination, they connect it to data memorization – training samples memorized by an NMT model are most likely to generate hallucinations when perturbed. For natural hallucinations, they connect them

to unrelated source and target pairs in the training data. Lee et al. (2018b) focus on the first type of hallucination (under input perturbations) and show that it is common in RNN-based NMT models. In addition, they discover that the average entropy of the encoder-decoder attention matrices associated with hallucinations under input perturbations is statistically different than those associated with input sentences that could not be perturbed to hallucination. However, it is unclear how their findings generalize to other NMT architectures (e.g. the Transformer models use multi-head attention rather than the dot-product one used in RNN-based models) and other types of hallucinations. In Chapter 4, we study internal model symptoms that are indicative of various types of hallucinations that are generalizable to various model architectures.

Detecting hallucinations is a challenging task. Most existing approaches to detecting hallucinations view the generation model as a black-box, by 1) training hallucination classifiers given the inputs and outputs alone on synthetic data constructed by heuristics (Santhanam et al., 2021; Zhou et al., 2021), or 2) using external semantic models such as question answering systems or natural language inference models to measure the faithfulness of model outputs (Durmus et al., 2020; Falke et al., 2019). These approaches ignore the signals from the generation model itself and could be highly biased by the heuristics used for synthetic data construction or the biases in the external semantic models trained for other purposes. In Chapter 4, we investigate varying types of glass-box features to detect hallucinations.

Detecting hallucinations in MT can also be seen as a particular case of MT quality estimation. A majority of the quality estimation work focuses on predicting the manually annotated direct assessment score for the perceived quality of a translation, which does not distinguish adequacy and fluency errors (Guzmán et al., 2019; Specia et al., 2020). More recently, a new task that aims to predict critical adequacy errors in MT was introduced in WMT

2021 (Specia et al., 2021). While hallucination is considered as a type of critical error, the error categories defined in this task do not distinguish hallucinations from other error types. In prior work, MT quality estimation is primarily addressed by: 1) black-box methods that estimate the output quality based on the source and output alone (Kim et al., 2017; Ranasinghe et al., 2020; Specia et al., 2009), and 2) glass-box methods based on features extracted from the NMT model itself (e.g. model probabilities, uncertainty quantification, and the entropy of the attention distribution) (Fomicheva et al., 2020; Rikters and Fishel, 2017; Yankovskaya et al., 2018). Black-box methods typically use resource-heavy deep neural networks trained on large amounts of in-domain labeled data. Our hallucination detection approach in Chapter 4 is more related to the glass-box methods that rely on internal model signals for quality estimation but differs in the type of signals used – our approach is based on the token contribution patterns extracted using a saliency-based interpretation method, which provide sharper features to characterize hallucinations than the ones used in existing quality estimation methods.

2.2 Integrating Inductive Biases in Text Generation Models

To address the aforementioned problems in NMT, we propose to integrate strong inductive biases in NMT models. Prior work has proposed various approaches to integrating stronger inductive biases in MT and more generally text generation models, which generate text freely or conditioned on an input (which can be a source language text for MT, a long segment of text for text summarization, or an image for image captioning). In this section, we introduce how inductive biases can be integrated in text generation models in the forms of multi-task learning objectives (Section 2.2.1), sequence-level training algorithms (Section 2.2.2), and more flexible

generation schemes (Section 2.2.3).

2.2.1 Multi-Task Learning

A common way to integrate stronger inductive biases into a model is through multi-task learning (Caruana, 1997), which is the set of techniques that trains a model on two or more tasks with some shared parameters (Ruder, 2017). Collobert et al. (2011); Hashimoto et al. (2017) showed that multi-task learning with tasks at various linguistic levels of morphology, syntax, and semantics improves the performance on each task. In the context of MT, multi-task learning has been used to inject linguistic knowledge using tasks such as part-of-speech tagging (Kiperwasser and Ballesteros, 2018; Niehues and Cho, 2017), named entity recognition (Niehues and Cho, 2017; Zaremoodi and Haffari, 2018), syntactic parsing (Eriguchi et al., 2017; Kiperwasser and Ballesteros, 2018; Luong et al., 2016), and semantic parsing (Zaremoodi and Haffari, 2018). These approaches lead to improved translation quality especially when the parallel data is limited (Zaremoodi and Haffari, 2018). However, they require annotated data on the auxiliary tasks, which is expensive to collect for truly low-resource languages.

For low-resource MT, a more promising direction is to design multi-task objectives to better leverage the parallel and monolingual data. We explore more effective use of parallel data by training NMT jointly on the traditional maximum likelihood objective and a novel training objective to improve translation quality (Chapter 3). To better leverage monolingual data, Luong et al. (2016) propose an unsupervised autoencoding objective that trains an NMT model to predict the next token in source and target monolingual texts. Semi-supervised MT methods such as Back-translation (Sennrich et al., 2016b) and Dual Learning (He et al., 2016) can also be viewed

as multi-task learning, as they train MT models jointly on the supervised and unsupervised learning objectives. Despite the improvements they have brought in practice, there is little understanding on what inductive biases are introduced by these objectives. In Chapter 5, we will introduce a theoretical framework to analyze what these objectives are optimizing toward and how they compare.

2.2.2 Sequence-Level Training for Text Generation Models

Another way to integrate stronger inductive biases in text generation models is through sequence-level training instead of the maximum likelihood training, which is typically used to train generation models by maximizing the likelihood of ground-truth sequences (Bahdanau et al., 2015; Sutskever et al., 2014). While simple and effective, maximum likelihood training suffers from the *exposure bias* problem (Ranzato et al., 2015): the model is only exposed to reference target sequences during training, but has to rely on its own predictions for the previous tokens in the target sequences at inference time. As a result, errors can accumulate along the generated sequence at inference time.

To address the problem, one needs to expose the model to its own predictions at training time through sequence-level training. To this end, Daumé et al. (2009) propose the Learning to Search approach with an iterative training process. At each training iteration, it first uses the current model to predict an output sequence up to the time step T. Then, at each time step t, it computes the cost-to-go of each possible token choice given the current prefix by completing the prefix using the oracle strategy. Finally, a new classifier is trained to minimize the cost-to-go. Chang et al. (2015) propose a similar algorithm that uses a mixture of the oracle and

model's strategy to compute the cost-to-go. However, both algorithms assume access to an oracle that predicts the optimal token given any prefix. While this may be easily computed in some applications, it is intractable to compute in natural text generation due to the large vocabulary and complicated cost functions.

For time series modeling, the Data as Demonstrator algorithm (Venkatraman et al., 2015) derives the oracle policy directly from the reference sequences assuming that they are aligned with the sampled sequences at each time step. Scheduled sampling algorithms (Bengio et al., 2015; Goyal et al., 2017) use the same strategy for text generation, even though the time-indexed alignment between reference and sampled sequences does not hold. Leblond et al. (2018) propose an approximation to the oracle policy by completing a predicted prefix with all possible reference suffixes and picking the reference suffix that yields the highest BLEU-1 score. However, they found that this approach performs well only when the prefix is close to the reference.

Another line of work explores reinforcement learning (RL) algorithms (Bahdanau et al., 2016; Sutton and Barto, 2018; Van Hasselt et al., 2016) to directly optimize a sentence-level reward given model generated prefixes instead of searching for the optimal token. Due to the discreteness of the token predictions and reward functions, we cannot directly back-propagate the gradients of the reward function to the model parameters. Techniques such as policy gradient (Williams, 1992) and actor-critic (Degris et al., 2012; Sutton and Barto, 2018) are thus required to find an unbiased estimation of the gradient to optimize the model. Due to the high variance of the gradient estimation, training with RL can be slow and unstable (Henderson et al., 2018; Wu et al., 2018). In Chapter 3, we will introduce a differentiable sampling algorithm that exposes machine translation models to their own predictions during training without requiring searching for the optimal token or training with RL.

2.2.3 Non-Monotonic and Non-Autoregressive Text Generation

The generation scheme is also a source of inductive bias in text generation models. For example, autoregressive models that generate text from left to right assume an autoregressive context dependency – each token prediction is conditioned only on the tokens on the left side (Cho et al., 2014a; Chorowski et al., 2015; Vinyals and Le, 2015). This generation scheme lacks the flexibility to meet the users' needs for constrained generation, infilling, fast decoding, etc. Non-monotonic and non-autoregressive models that break the left-to-right generation order have been proposed. One line of research focusing on increasing the decoding speed propose fully non-autoregressive models that generate all output tokens in a single step (Gu et al., 2018; Ma et al., 2019; van den Oord et al., 2018). However, their output quality suffers due to the large decoding space and strong independence assumptions between target tokens (Ma et al., 2019; Wang et al., 2019). Other works propose to generate the output sequence through multiple iterations with more flexibility in the generation order and the number of tokens to generate at each iteration. As an example, Stern et al. (2019) introduce the Insertion Transformer that generates text through iterative insertion – at each iteration, the model predicts the token to insert and the insertion location relative to the current sequence. This generation scheme is flexible enough to accommodate arbitrary generation order, including left-to-right, uniform (i.e. no preference over any particular order), and balanced binary tree in which the centermost token in a span is always inserted first. This framework also allows for fast decoding by inserting multiple tokens at each iteration (i.e. parallel decoding). They show that Insertion Transformer with parallel decoding achieves competitive MT quality to autoregressive baselines while using substantially fewer iterations during decoding. Ghazvininejad et al. (2019) propose a Mask-

Predict generation scheme where the model first predicts the length of the output sequence and all output tokens, and then it iteratively edits the sequence by masking a subset of the tokens and re-predict the masked tokens. Compared with the Insertion Transformer, this increases the flexibility at inference time as it allows for corrections to the previously generated tokens. But it has its limitation: its iterative refinement process is bounded by a fixed length predicted at the first iteration. To break this limit, Gu et al. (2019) propose Levenshtein Transformer, which iteratively edits the output sequence through insertion and deletion operations so that the length of the output sequence can be changed dynamically through iterations. However, it is still limited in flexibility due to the entangled lexical choice and reordering decisions, i.e. the relative positions of previously generated tokens cannot be changed unless they are deleted. In Chapter 6, we will introduce the EDITOR model with a novel reposition operation that can dynamically adjust the positions of previously generated tokens.

2.3 Interpretation of Natural Language Processing Models

Despite the recent success of deep neural models on a wide range of natural language processing tasks, these models still make egregious errors (Feng et al., 2018; Jia and Liang, 2017). To better identify and understand such errors, it is important to understand why models make the predictions they do. This is a challenging problem due to the large number of input features consumed by a neural network, huge number of hidden units, and complicated structure (e.g. attention). Existing interpretation methods can be categorized into three categories: 1) model probing that inspect the model's representations for certain properties by training a classifier to perform linguistically-motivated probing tasks from the frozen representations (Belinkov et al.,

2017; Liu et al., 2019b; Tenney et al., 2019), 2) example-specific interpretation such as saliency maps (Simonyan et al., 2013) and input perturbations (Feng et al., 2018; Li et al., 2016b) that aims to understand which parts of an input lead to a prediction, and 3) generative explanation that trains a model to generate natural language explanations for its predictions (Liu et al., 2019a). It remains an open question how to properly evaluate and choose between interpretation methods for a specific use case. As Ribeiro et al. (2016) point out, there is a tradeoff between interpretability and fidelity in the design of interpretation methods. For example, generative explanation methods generally have higher interpretability than gradient-based methods, but it may not faithfully explain how the prediction is actually formed. In Chapter 4, we apply interpretation methods to better understand how hallucinations are generated by NMT models and find model signals to detect them. Thus we focus on interpretation methods with high fidelity, which include the perturbation and saliency-based methods to measure the importance of each input unit to a prediction (Bach et al., 2015; Ding et al., 2019; Li et al., 2016a; Simonyan et al., 2013). While other saliency-based methods measure an abstract quantity reflecting the importance of each input feature by the partial derivative of the prediction with regard to each input unit (Simonyan et al., 2013), Layerwise Relevance Propagation (LRP) (Bach et al., 2015) measures the proportional contribution of each input unit, which makes it well suited to contrast the model's behavior on different samples. Furthermore, LRP does not require neural activations to be differentiable and smooth, and can be applied to a wide range of model architectures. It has been used to analyze various NMT model architectures including RNN (Ding et al., 2017) and Transformer (Voita et al., 2021). In Chapter 4, we apply this technique to analyze counterfactual hallucination samples inspired by perturbation-based methods (Ebrahimi et al., 2018; Feng et al., 2018; Li et al., 2016b), and show that the insights generalize to hallucinations generated on naturally occurring text.

Chapter 3: Mitigating Exposure Bias by Strengthening Inductive Biases

3.1 Introduction

We start by showing that integrating inductive biases in the form of training objectives helps mitigate undesirable biases in NMT models that hamper effective learning from limited training data. In this work, we focus on addressing a well-known undesirable bias from the maximum likelihood objective typically used in NMT training (Bahdanau et al., 2015; Sutskever et al., 2014). While simple and effective, this objective suffers from the exposure bias problem (Ranzato et al., 2015): the model is only exposed to the reference prefixes during training, but has to rely on its predicted prefixes during inference. As a result, errors can accumulate along the generated sequence at inference time.

This is a well-known issue in sequential decision making (Cohen and Carvalho, 2005; Kääriäinen and Langford, 2006; Langford and Zadrozny, 2005, i.a.) and it has been addressed in past work by incorporating the previous decoding choices into the training scheme, using imitation learning (Bengio et al., 2015; Daumé et al., 2009; Leblond et al., 2018; Ross et al., 2011) and reinforcement learning (Bahdanau et al., 2016; Ranzato et al., 2015) techniques. In this chapter, we focus on a simple and computationally inexpensive family of approaches, known as Data as Demonstrator (Venkatraman et al., 2015) and scheduled sampling (Bengio et al., 2015; Goyal et al., 2017). The algorithms use a stochastic mixture of the reference words and

model predictions with an annealing schedule controlling the mixture probability. Despite their empirical success in various sequence prediction tasks, they are based on an assumption that does not hold for machine translation: they assume that words in the reference translations and in sampled sequences are aligned at each time step, which results in weak and sometimes misleading training signals.

We introduce a differentiable sampling algorithm that exposes machine translation models to their own predictions during training, and allows for differences in word order when comparing model outputs with reference translations. We compute the probability that the reference can be aligned with the sampled output using a soft alignment predicted based on the model states, so that the model will not be punished too severely for producing hypotheses that deviate from the reference, as long as the hypotheses can still be aligned with the reference.

Experiments on three IWSLT tasks (German-English, English-German and Vietnamese-English) show that our approach significantly improves BLEU compared to both maximum likelihood and scheduled sampling baselines. We also provide evidence that our approach addresses exposure bias by decoding with varying beam sizes, and show that our approach is simpler to train than scheduled sampling as it requires no annealing schedule.

3.2 Soft-Aligned Maximum Likelihood Objective

Our approach is designed to optimize the standard sequence-to-sequence model for translating a source sentence $\boldsymbol{x}$ into a target sentence $\boldsymbol{y}$ (Bahdanau et al., 2015). This model

computes the probability of $\boldsymbol{y}$ given $\boldsymbol{x}$ as:

$$P(\boldsymbol{y}\,|\,\boldsymbol{x}) = \prod_{t=1}^{T} p(y_t\,|\,\boldsymbol{y}_{<t}, \boldsymbol{x}; \theta) \tag{3.1}$$

where θ represents the model parameters. Given $\boldsymbol{x}$, the model first produces a sequence of hidden representations $\boldsymbol{h}_{1\ldots T}$: $\boldsymbol{h}_t = f(\boldsymbol{y}_{<t}, \boldsymbol{x})$, where T is the length of $\boldsymbol{y}$, and f is usually an encoder-decoder network. At each time step t, the hidden representation $\boldsymbol{h}_t$ is fed to a linear projection layer $\boldsymbol{s}_t = \boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}$ to obtain a vector of scores $\boldsymbol{s}_t$ over all possible words in the vocabulary $\mathscr{V}$. Scores are then turned into a conditional probability distribution: $p(\cdot\,|\,\boldsymbol{y}_{<t}, \boldsymbol{x}; \theta) = \text{softmax}(\boldsymbol{s}_t)$.

The traditional maximum likelihood (ML) objective maximizes the log-likelihood of the training data $\mathscr{D} \equiv \{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}_{n=1}^{N}$ consisting of N pairs of source and target sentences:

$$\mathscr{J}_{ML}(\theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(y_t^{(n)}\,|\,\boldsymbol{y}_{<t}^{(n)}, \boldsymbol{x}^{(n)}; \theta) \tag{3.2}$$

At test time, prefixes $\boldsymbol{y}_{<t}$ are subsequences generated by the model and therefore contain errors. By contrast, in ML training, prefixes $\boldsymbol{y}_{<t}$ are subsequences of reference translations. As a result, the model is never exposed to its own errors during training and errors accumulate at test time. This mismatch is known as the exposure bias problem (Ranzato et al., 2015).

3.2.1 Limitations in Scheduled Sampling

Bengio et al. (2015) introduced the scheduled sampling algorithm to address exposure bias. Scheduled sampling gradually replaces the reference words with sampled model predictions in the prefix used at training time. An annealing schedule controls the probability of using reference

words vs. model predictions. The training objective remains the same as the ML objective, except for the nature of the prefix $\hat{\boldsymbol{y}}_{<t}$, which contains a mixture of reference and predicted words:

$$\mathcal{J}_{SS}(\theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log p(y_t^{(n)} \mid \hat{\boldsymbol{y}}_{<t}^{(n)}, \boldsymbol{x}^{(n)}; \theta) \tag{3.3}$$

Despite the empirical success of scheduled sampling, one limitation is that the discontinuity of the argmax operation makes it impossible to penalize errors made in previous steps, which can lead to slow and unstable training. We address this issue using a continuous relaxation to the greedy search and sampling process, similarly to Goyal et al. (2017), which we describe in Section 3.2.2.

Another limitation of scheduled sampling is that it incorrectly assumes that the reference and predicted sequence are aligned by time indices which introduces additional noise to the training signal.[1] We address this problem with a novel differentiable sampling algorithm with an alignment based objective called soft aligned maximum likelihood (SAML). It is used in combination with maximum likelihood to define our training objective $\mathcal{J} = \mathcal{J}_{ML} + \mathcal{J}_{SAML}$, where $\mathcal{J}_{ML}$ is computed based on reference translations, and $\mathcal{J}_{SAML}$ is computed based on sampled translations of the same input sentences. We define $\mathcal{J}_{SAML}$ in Section 3.2.3.

3.2.2 Differentiable Sampling

To backpropagate errors made in the previous decoding steps, we use a continuous relaxation of the discrete sampling operation similar to Goyal et al. (2017), except that we use the Straight-Through (ST) Gumbel-Softmax estimator (Bengio et al., 2013; Jang et al., 2017) instead

[1] https://nlpers.blogspot.com/2016/03/a-dagger-by-any-other-name-scheduled.html

of Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2014) to better simulate the scenario at inference time.[2]

The Gumbel-Softmax is derived from the Gumbel-Max trick (Maddison et al., 2014), an algorithm for sampling one-hot vector $z \in \mathbb{R}^k$ from a categorical distribution $(p_1, ..., p_k)$:

$$z = \text{one-hot}(\arg\max_i(\log p_i + \beta g_i)) \tag{3.4}$$

where g_i is the Gumbel noise drawn i.i.d from Gumbel$(0,1)$[3], and β is a hyperparameter controlling the scale of the noise. Here, the trick is used to approximate the discontinuous argmax function with the differentiable softmax:

$$\tilde{z} = \text{softmax}((\log p_i + \beta g_i)/\tau) \tag{3.5}$$

where τ is the temperature parameter. As τ diminishes to zero, $\tilde{z}$ becomes the same as one-hot sample z.

The Straight-Through Gumbel-Softmax maintains the differentiability of the Gumbel-Softmax estimator while allowing for discrete sampling by taking different paths in the forward and backward pass. It uses argmax to get the one-hot sample z in the forward pass, but uses its continuous approximation $\tilde{z}$ in the backward pass. While ST estimators are biased, they have been shown to work well in latent tree learning (Choi et al., 2018) and semi-supervised machine translation (Niu et al., 2019).

[2]The Straight-Through estimator consistently outperforms the Gumbel-Softmax in preliminary experiments.
[3]$g_i = -\log(-\log(u_i))$ and $u_i \sim \text{Uniform}(0,1)$.

3.2.3 Soft Aligned Maximum Likelihood

The soft aligned maximum likelihood (SAML) is defined as the probability that the reference can be aligned with the sampled output using a soft alignment predicted by the model:

$$P_{SAML}(\boldsymbol{y}\,|\,\boldsymbol{x}) = \prod_{t=1}^{T}\sum_{j=1}^{T'} a_{tj}\cdot p(y_t\,|\,\tilde{\boldsymbol{y}}_{<j},\boldsymbol{x};\theta) \tag{3.6}$$

where T is the length of the reference sequence, T' is the length of the sampled sequence, a_{tj} is the predicted soft alignment between the reference word y_t and sampled prefix $\tilde{\boldsymbol{y}}_{<j}$.

Training with the SAML objective consists in maximizing:

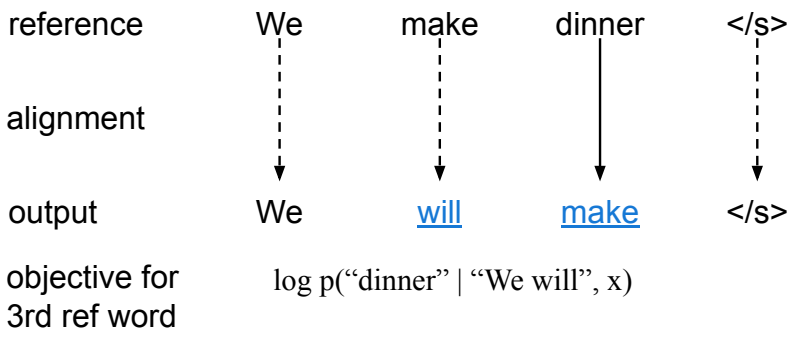
$$\mathscr{J}_{SAML}(\theta) = \sum_{n=1}^{N} \log P_{SAML}(\boldsymbol{y}^{(n)}\,|\,\boldsymbol{x}^{(n)}) \tag{3.7}$$

The conditional probability of the next word $p(y_t\,|\,\tilde{\boldsymbol{y}}_{<j},\boldsymbol{x};\theta)$ is computed as follows:

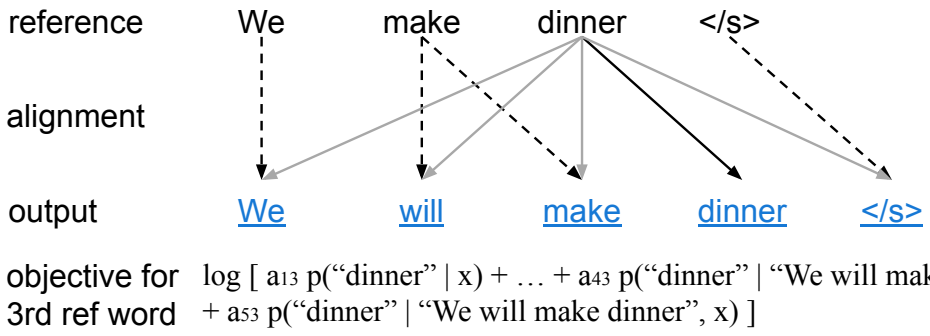
$$p(\cdot\,|\,\tilde{\boldsymbol{y}}_{<j},\boldsymbol{x};\theta) = \text{softmax}(\boldsymbol{W}\tilde{\boldsymbol{h}}_j + \boldsymbol{b}) \tag{3.8}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are model parameters. $\tilde{\boldsymbol{h}}_j$ is the hidden representation at step j conditioned on the source sequence $\boldsymbol{x}$ and the preceding words $\tilde{\boldsymbol{y}}_{<j}$ sampled from the model distribution using differentiable sampling:

$$\tilde{\boldsymbol{h}}_j = f(\tilde{\boldsymbol{y}}_{<j},\boldsymbol{x}) \tag{3.9}$$



(a) Scheduled Sampling Objective



(b) SAML Objective

Figure 3.1: Difference between objectives used in scheduled sampling (left) and our approach (right), when computing the contribution to the objective of the reference word “dinner”. The schedule sampling hypothesis uses a mixture of the reference (black) and sampled (blue underlined) words, while the entire hypothesis sequence is sampled in our approach.

We compute the soft alignment a_{tj} between y_t and $\tilde{\mathbf{y}}_{<j}$ based on the model's hidden states:

$$a_{tj} = \frac{\exp(score(\tilde{\boldsymbol{h}}_j, \boldsymbol{e}_{y_t}))}{\sum_{i=1}^{T'} \exp(score(\tilde{\boldsymbol{h}}_i, \boldsymbol{e}_{y_t}))} \tag{3.10}$$

where $\boldsymbol{e}_{y_t}$ is the embedding of the reference word y_t. The *score* function captures the similarity between the hidden state $\tilde{\boldsymbol{h}}_j$ and the embedding $\boldsymbol{e}_{y_t}$. We use the dot product here as it does not introduce additional parameters:

$$score(\boldsymbol{h}, \boldsymbol{e}) = \boldsymbol{h}^\top \boldsymbol{e} \tag{3.11}$$

Figure 3.1 illustrates how the resulting objective differs from scheduled sampling: (1) it is computed over sampled sequences as opposed to sequences that contain a mixture of sampled

and reference words, and (2) each reference word is soft-aligned to the sampled sequence.

3.3 Experiments

3.3.1 Experimental Settings

Data We evaluate our approach on IWSLT 2014 German-English (de-en) as prior work (Goyal et al., 2017), as well as two additional tasks: IWSLT 2014 English-German (en-de) and IWSLT 2015 Vietnamese-English (vi-en). For de-en and en-de, we follow the preprocessing steps in Ranzato et al. (2015). For vi-en, we use the data preprocessed by Luong and Manning (2015), with test2012 for validation and test2013 for testing. Table 3.1 summarizes the data statistics.

Task	sentences (K)			vocab (K)	
	train	dev	test	src	tgt
de-en	153.3	7.0	6.8	113.5	53.3
vi-en	121.3	1.5	1.3	23.9	50.0

Table 3.1: We evaluate on two translation tasks.

Setup Our translation models are attentional RNNs (Bahdanau et al., 2015) built on Sockeye (Hieber et al., 2017). We use bi-directional LSTM encoder and single-layer LSTM decoder with 256 hidden units, embeddings of size 256, and multilayer perceptron attention with a layer size of 256. We apply layer normalization (Ba et al., 2016) and label smoothing (0.1). We add dropout to embeddings (0.1) and decoder hidden states (0.2). For ST Gumbel-Softmax, we use temperature $\gamma = 1$ and noise scale $\beta = 0.5$. The decoding beam size is 5 unless stated otherwise. We train the models using the Adam optimizer (Kingma and Ba, 2015) with a batch size of 1024 words. We checkpoint models every 1000 updates. The initial learning rate is 0.0002, and it is reduced by 30% after 4 checkpoints without validation perplexity

improvement. Training stops after 12 checkpoints without improvement. For training efficiency, we first pre-train a baseline model for each task using only $\mathscr{J}_{ML}$ and fine-tune it using different approaches. In the fine-tuning phase, we inherit all settings except that we initialize the learning rate to 0.00002 and set the minimum number of checkpoints before early stopping to 24. We fine-tune each randomly seeded model independently.

Baselines We compare our model against three baselines: (1) a standard baseline trained with the ML objective, and models fine-tuned with (2) scheduled sampling (**SS**) (Bengio et al., 2015) and (3) differentiable scheduled sampling (**DSS**) (Goyal et al., 2017). In SS and DSS, the probability of using reference words ε_s is annealed using inverse sigmoid decay (Bengio et al., 2015): $\varepsilon_s = k/(k + \exp(i/k))$ at the i-th checkpoint with $k = 10$.

3.3.2 Results

Table 3.2 shows that the SAML improves over the ML baseline by +0.5 BLEU on de-en, +0.7 BLEU on en-de, and +1.0 BLEU on vi-en task. In addition, SAML consistently improves over both the scheduled sampling and differentiable scheduled sampling on all tasks. All improvements are significant except for SAML versus scheduled sampling on vi-en.[4] Interestingly, differentiable scheduled sampling performs no better than scheduled sampling in our experiments, unlike in Goyal et al. (2017).

Unlike scheduled sampling, our approach does not require an annealing schedule, and it is therefore simpler to train. We verify that the annealing schedule is needed in scheduled sampling by training a contrastive model with the same objective as scheduled sampling, but without

[4]The statistical test is based on independent student's t-test with Bonferroni Correction ($p < 0.05$) over 5 runs.

Method	Anneal	de-en	en-de	vi-en
Baseline	No	27.41$_{\pm 0.26}$	22.64$_{\pm 0.13}$	23.59$_{\pm 0.13}$
+SS	Yes	27.47$_{\pm 0.28}$	22.56$_{\pm 0.17}$	23.97$_{\pm 0.39}$
+DSS	Yes	27.30$_{\pm 0.24}$	22.47$_{\pm 0.20}$	23.68$_{\pm 0.35}$
+SS	No	22.91$_{\pm 0.21}$	17.78$_{\pm 0.20}$	19.57$_{\pm 0.19}$
+SAML	No	**27.94**$_{\pm 0.12}$	**23.30**$_{\pm 0.19}$	**24.60**$_{\pm 0.35}$

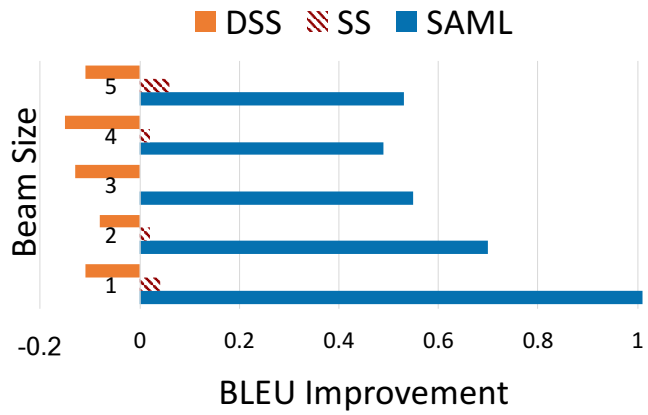
Table 3.2: BLEU scores of our approach (SAML) and three baselines including the maximum likelihood (ML) baseline, scheduled sampling (SS), and differentiable scheduled sampling (DSS). The *Anneal* column indicates whether the sampling rate is annealed. For each task, we report the mean and standard deviation over 5 runs with different random seeds. SAML achieves the best BLEU scores and is simpler to train than SS and DSS, as it requires no annealing schedule.

annealing schedule (Table 3.2). We set the sampling rate to 0.5. The contrastive model hurts BLEU scores by at least 4.0 points compared to both the ML baseline and models fine-tuned with scheduled sampling, confirming that scheduled sampling needs the annealing schedule to work well.

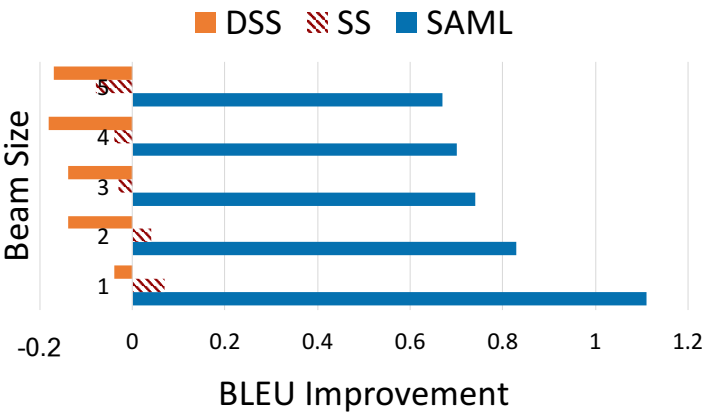
We further examine the performance gain of different approaches over the baseline with varying beam sizes (Figure 3.2). Our approach yields larger BLEU improvements when decoding with greedy search and smaller beams, while there is no clear pattern for scheduled sampling models. These results support the hypothesis that our approach mitigates exposure bias, as it yields bigger improvements in settings where systems have fewer opportunities to recover from early errors.

3.3.3 Limitations

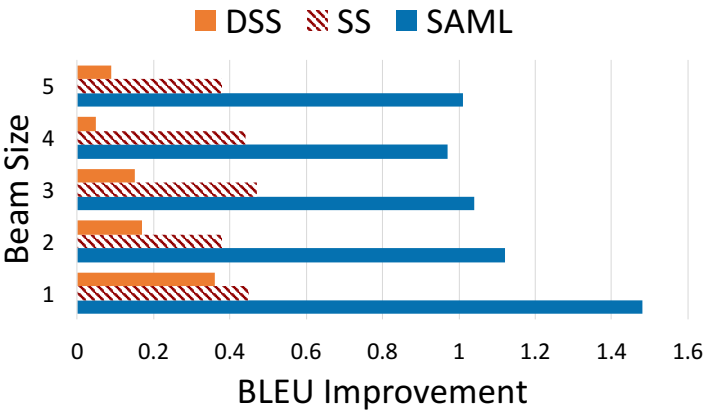
The findings of this work should be interpreted with several limitations in mind. First, we only evaluate our approach on three translation tasks, so it remains an open question how this approach would perform on other language pairs with different linguistic properties. Second, we primarily rely on BLEU as the evaluation metric for the empirical study, so it remains to be



(a) de-en



(b) en-de



(c) vi-en

Figure 3.2: Improvements from our method (SAML), scheduled sampling (SS), and differentiable scheduled sampling (DSS) over the maximum likelihood (ML) baseline when decoding with varying beam sizes (average of 5 runs). The SAML model consistently yields the largest improvements with smaller beams.

explored how the improvements brought by our approach would be perceived by humans.

3.4 Summary

We introduced a differentiable sampling algorithm which addresses the exposure bias problem by exposing an NMT model to its own predictions during training and comparing them to reference sequences flexibly to back-propagate reliable error signals. By soft aligning reference and sampled sequences, our approach consistently improves BLEU over maximum likelihood and scheduled sampling baselines on three translation tasks, with larger improvements when using greedy search and smaller beam sizes. Our approach is also simple to train, as it does not require any sampling schedule.

This demonstrates that integrating inductive biases in the form of training objectives helps mitigate exposure bias, a well-known undesirable bias in NMT models. In the next chapter, we turn to study the spurious biases that lead to hallucinations – a well-known but less studied problem in MT where the model produces a translation that is unrelated to the source input.

Chapter 4: Identifying Spurious Biases That Lead to Hallucinations

4.1 Introduction

Apart from exposure bias, another type of undesirable bias that can hurt the models' generalization ability is the spurious bias – prediction rules that work well for a subset of the training samples but do not generalize to other samples. In this chapter, we study the spurious biases that lead an egregious type of MT error, namely "detached hallucinations" (Raunak et al., 2021), where the output is completely detached from the source. Such errors not only reveal fundamental limitations in the generalization abilities of current models, but also risk misleading users and undermining trust (Bender et al., 2021). Yet, we lack a systematic understanding of the conditions where hallucinations arise, as hallucinations occur infrequently among translations of naturally occurring text. As a workaround, prior work has largely focused on black-box detection methods which train neural classifiers on synthetic data constructed by heuristics (Falke et al., 2019; Zhou et al., 2021), and on studying hallucinations given artificially perturbed inputs (Lee et al., 2018b).

In this chapter, we address the problem by first identifying the internal model symptoms that characterize hallucinations given artificial inputs and then testing the discovered symptoms on translations of natural texts. Specifically, we study hallucinations in Neural Machine Translation (NMT) using two types of interpretability techniques. We use saliency analysis

methods (Bach et al., 2015; Voita et al., 2021) to compare the relative contribution of various tokens to the hallucinated vs. non-hallucinated outputs generated by diverse adversarial perturbations in the inputs (Table 4.1) inspired by Lee et al. (2018b); Raunak et al. (2021). Results surprisingly show that source contribution patterns are stronger hallucination indicators than the relative contribution of the source and target, as had been previously hypothesized (Voita et al., 2021). We discover two distinctive source contribution patterns, including 1) concentrated contribution from a small subset of source tokens, and 2) the staticity of the source contribution distribution along the generation steps (Section 4.3).

We further show that the symptoms identified generalize to hallucinations on natural inputs by using them to design a lightweight hallucination classifier (Section 4.4) that we evaluate on manually annotated hallucinations from English-Chinese and German-English NMT (Table 4.1). Our study shows that our introspection-based detection model largely outperforms model-free baselines and the classifier based on quality estimation scores. Furthermore, it is more accurate and robust to domain shift than black-box detectors based on large pre-trained models (Section 4.5).

Finally, we provide evidence that the hallucination symptoms are connected with the spurious biases learned on certain training samples. Specifically, we study the training samples that contain similar target text segments to the hallucinated outputs and find that these training samples exhibit similar source contribution patterns as the hallucination samples at inference time (Section 4.6).

Before presenting these two studies, we review current findings about the conditions in which hallucinations arise and formulate three hypotheses capturing potential hallucination symptoms.

Counterfactual hallucination from perturbation	
Source	Republicans Abroad are not running a similar election, nor will they have delegates at the convention. Recent elections have emphasized the value of each vote.
Good NMT	国外的共和党没有举行类似的选举，也没有代表参加大会。最近的选举强调了每次投票的价值。
Perturbed Source	Repulicans Abroad ar not runing a simila election, nor will they have delegates at the convention. Recent elections have emphasized the value o each vote.
Hallucination	大耳朵评论管理人员有权保留或删除其管辖评论中的任意内容。 *Gloss: The big ear comments that administrators have the right to retain or delete any content in the comments under their jurisdiction.*
Natural hallucination	
Source	DAS GRUNDRECHT JEDES EINZELNEN AUF FREIE WAHL DES BERUFS, DER AUSBILDUNGSSTÄTTE SOWIE DES AUSBILDUNGS - UND BESCHÄFTIGUNGSORTS MUSS GEWAHRT BLEIBEN. *Gloss: The fundamental right of every individual to freely choose their profession, their training institution and their employment place must remain guaranteed.*
Hallucination	THE PRIVACY OF ANY OTHER CLAIM, EXTRAINING STANDARDS, EXTRAINING OR EMPLOYMENT OR EMPLOYMENT WILL BE LIABLE.

Table 4.1: Contrasting counterfactual English-Chinese hallucinations derived from source perturbations (top) with a natural hallucination produced by a German-English NMT model (bottom).

4.2 Hallucinations: Definition and Hypotheses

The term "hallucinations" has varying definitions in MT and natural language generation. We adopt the most widely used one, which refers to output text that is unfaithful to the input (Ji et al., 2022; Maynez et al., 2020a; Xiao and Wang, 2021b; Zhou et al., 2021).[1] Different from the previous work that aims to detect partial hallucinations at the token level (Zhou et al., 2021), we focus on **detached hallucinations** where a major part of the output is unfaithful to the input, as these represent severe errors, as illustrated in Table 4.1.

Prior work on understanding the conditions that lead to hallucinations has focused on training conditions and data noise (Ji et al., 2022). For MT, Raunak et al. (2021) show that

[1] Others include fluency criteria as part of the definition (Martindale et al., 2019; Wang and Sennrich, 2020).

hallucinations under perturbed inputs are caused by training samples in the long tail that tend to be memorized by Transformer models, while natural hallucinations given unperturbed inputs can be linked to corpus-level noise. Wang and Sennrich (2020) establish a link between MT hallucinations under domain shift and exposure bias by showing that Minimum Risk Training, a training objective which addresses exposure bias, can reduce the frequency of hallucinations. However, these insights do not yet provide practical strategies for handling MT hallucinations.

A complementary approach to diagnosing hallucinations is to identify their symptoms via model introspection at inference time. However, there lacks a systematic study of hallucinations from the model's internal perspective. Previous works are either limited to an interpretation method that is tied to an outdated model architecture (Lee et al., 2018b) or to pseudo-hallucinations (Voita et al., 2021). In this work, we propose to shed light on the decoding behavior of hallucinations on both artificially perturbed and natural inputs through model introspection based on Layerwise Relevance Propagation (LRP) (Bach et al., 2015), which is applicable to a wide range of neural model architectures. We focus on MT tasks with the widely used Transformer model (Vaswani et al., 2017), and examine existing and new hypotheses for how hallucinations are produced. These hypotheses share the intuition that anomalous patterns of contributions from source tokens are indicative of hallucinations, but operationalize it differently.

The **Low Source Contribution Hypothesis** introduced by Voita et al. (2021) states that hallucinations occur when NMT overly relies on the target context over the source. They test the hypothesis by inspecting the relative source and target contributions to NMT predictions on Transformer models using LRP. However, their study is limited to pseudo-hallucinations produced by force decoding with random target prefixes. This work will test this hypothesis on actual hallucinations generated by NMT models.

The **Local Source Contribution Hypothesis** introduced by Lee et al. (2018b) states that hallucinations occur when NMT model overly relies on a small subset of source tokens across all generation steps. They test it by visualizing the dot-product attention in RNN models, but it is unclear whether these findings generalize to other model architectures. In addition, they only study hallucinations caused by random token insertion. This work will test this hypothesis on hallucinations under various types of source perturbations as well as on natural inputs, and will rely on LRP to quantify token contributions more precisely than with attention.

Inspired by the previous observations on attention matrices that an NMT model attends repeatedly to the same source tokens throughout inference when it hallucinates (Berard et al., 2019b; Lee et al., 2018b) or generates a low-quality translation (Rikters and Fishel, 2017),[2] we formalize this observation as the **Static Source Contribution Hypothesis** – the distribution of source contributions remains static along inference steps when an NMT model hallucinates. Unlike the Low Source Contribution Hypothesis, this hypothesis exclusively relies on the source and does not make any assumption about relative source versus target contributions. Unlike the Local Source Contribution Hypothesis, this hypothesis is agnostic to the proportion of source tokens contributing to a translation.

In this work, we evaluate in a controlled fashion how well each hypothesis explains detached hallucinations, first on artificially perturbed samples that let us contrast hallucinated vs. non-hallucinated outputs in controlled settings (Section 4.3) and second, on natural source inputs that let us test the generalizability of these hypotheses when they are used to automatically detect hallucinations in more realistic settings (Section 4.5).

[2] Although they specifically highlight the static attention to the EOS or full-stop tokens.

4.3 Study of Hallucinations under Perturbations via Model Introspection

Hallucinations are typically rare and difficult to identify in natural datasets. To test the aforementioned hypotheses at scale, we first exploit the fact that source perturbations exacerbate NMT hallucinations (Lee et al., 2018b; Raunak et al., 2021). We construct a perturbation-based counterfactual hallucination dataset on English→Chinese. by automatically identifying hallucinated NMT translations given perturbed source inputs and contrast them with the NMT translations of the original source (Section 4.3.1). This dataset lets us directly test the three hypotheses by computing the relative token contributions to the model's predictions using LRP (Section 4.3.2), and conduct a controlled comparison of patterns on the original and hallucinated samples (Section 4.3.4).

4.3.1 Perturbation-based Hallucination Data

To construct the dataset, we randomly select $50k$ seed sentence pairs to perturb from the NMT training corpora, and then we apply the following perturbations on the source sentences:[3]

- We randomly misspell words by deleting characters with a probability of 0.1, as Karpukhin et al. (2019) show that a few misspellings can lead to egregious errors in the output.
- We randomly title-case words with a probability of 0.1, as Berard et al. (2019a) find that this often leads to severe output errors.
- We insert a random token at the beginning of the source sentence, as Lee et al. (2018b); Raunak et al. (2021) find it a reliable trigger of hallucinations. The inserted token is chosen

[3]For better contrastive analysis, we select samples with source length of $n = 30$ and clip the output length by $T = 15$.

from 100 most frequent, 100 least frequent, mid-frequency tokens (randomly sampled 100 tokens from the remaining tokens), and punctuations.

Inspired by Lee et al. (2018b), we then identify hallucinations using heuristics that compare the translations from the original and perturbed sources. We select samples whose original NMT translations y' are of reasonable quality compared to the reference y (i.e. $bleu(y, y') > 0.3$).[4] The translation of a perturbed source sentence $\tilde{y}$ is identified as a hallucination if it is very different from the translation of the original source (i.e. $bleu(y', \tilde{y}) < 0.03$) and is not a copy of the perturbed source $\tilde{x}$ (i.e. $bleu(\tilde{x}, \tilde{y}) < 0.5$).[5]

This results in 623, 270, and 1307 contrastive pairs of the original (non-hallucinated) and hallucinated translations under misspelling, title-casing, and insertion perturbations, respectively.

We further divide the contrastive pairs into degenerated and non-degenerated hallucinations. Degenerated hallucinations are "bland, incoherent, or get stuck in repetitive loops" (Holtzman et al., 2019), i.e. hallucinated translations that contain 3 more repetitive n-grams than the source are identified as degenerated hallucinations, while the non-degenerated group contains relatively fluent but hallucinated translations.

4.3.2 Measuring Relative Token Contributions

We test the three source contribution hypotheses described in Section 4.2 on the resulting dataset by contrasting the contributions of relevant tokens to the generation of a hallucinated versus a non-hallucinated translation using LRP (Bach et al., 2015). LRP decomposes the prediction of a neural model (originally designed for convolutional neural networks in computer

[4]Following Lee et al. (2018b), we set the weights for unigrams and bigrams to 0.6 and 0.4 and disregard other n-grams when computing sentence-level BLEU.

[5]The BLEU thresholds are selected based on manual inspection of the translation outputs.

vision) computed over an input instance into relevance scores for input dimensions. Specifically, LRP decomposes a neural model into several layers of computation and measures the relative influence score $R_i^{(l)}$ for input neuron i at layer l. Different from other interpretation methods that measure the absolute influence of each input dimension (Alvarez-Melis and Jaakkola, 2017; He et al., 2019; Ma et al., 2018), LRP adopts the principal that the relative influence $R_i^{(l)}$ from all neurons at each layer should sum up to a constant:

$$\sum_i R_i^{(1)} = \sum_i R_i^{(2)} = ... = \sum_i R_i^{(L)} = C \tag{4.1}$$

To back-propagate the influence scores from the last layer to the first layer (i.e. the input layer), we need to decompose the relevance score $R_j^{(l+1)}$ of a neuron j at layer $l+1$ into messages $R_{i\leftarrow j}^{(l,l+1)}$ sent from the neuron j at layer $l+1$ to each input neuron i at layer l under the following rules:

$$R_{i\leftarrow j}^{(l,l+1)} = v_{ij} R_j^{(l+1)}, \sum_i v_{ij} = 1 \tag{4.2}$$

There exist several versions of LRP, including LRP-ε, LRP-$\alpha\beta$, and LRP-γ, which compute v_{ij} differently (Bach et al., 2015; Binder et al., 2016; Montavon et al., 2019). Following Voita et al. (2021), we use LRP-$\alpha\beta$ (Bach et al., 2015; Binder et al., 2016), which defines v_{ij} such that the relevance scores are positive at each step. Specifically, in the simplest case of linear layers with non-linear activation functions:

$$u_j^{(l+1)} = g(z_j), z_j = \sum_i z_{ij} + b_j, z_{ij} = w_{ij} u_i^{(l)} \tag{4.3}$$

where $u_i^{(l)}$ is the i-th neuron at layer l, w_{ij} is the weight connecting the neurons $u_i^{(l)}$ and $u_j^{(l+1)}$, b_j

is a bias term, and g is a non-linear activation function. Next, the $\alpha\beta$ rule considers the positive and negative contributions separately:

$$z_{ij}^{+} = \max(z_{ij}, 0), b_j^{+} = \max(b_j, 0)$$

$$z_{ij}^{-} = \min(z_{ij}, 0), b_j^{-} = \min(b_j, 0)$$

and defines v_{ij} by the following equation:

$$v_{ij} = \alpha \cdot \frac{z_{ij}^{+}}{\sum_i z_{ij}^{+} + b_j^{+}} + \beta \cdot \frac{z_{ij}^{-}}{\sum_i z_{ij}^{-} + b_j^{-}} \tag{4.4}$$

Following Voita et al. (2021), we use $\alpha = 1, \beta = 0$. This rule is directly applicable to linear, convolutional, maxpooling, and feed-forward layers. To back-propagate relevance scores through attention layers in the Transformer encoder-decoder model (Vaswani et al., 2017), we follow the propagation rules in Voita et al. (2021), where the weighting v_{ij} is obtained by performing a first order Taylor expansion of each neuron $u_j^{(l+1)}$.

In the context of NMT, LRP ensures that, at each generation step t, the sum of contributions $R_t(x_i)$ and $R_t(y_j)$ from source tokens x_i and target prefix tokens y_j remains equal:

$$\forall t, \sum_i R_t(x_i) + \sum_{j<t} R_t(y_j) = 1 \tag{4.5}$$

We further define normalized source contribution $\bar{R}(x_i)$ at source position i averaged over all generation steps t as:

$$\bar{R}(x_i) = \frac{1}{T} \sum_t^T \frac{n \cdot R_t(x_i)}{\sum_i^n R_t(x_i)} \tag{4.6}$$

where n is the length of each source sequence and T is the length of the output sequence.

We then test the aforementioned hypotheses based on the distribution of relative token contributions and compare it with the attention matrix.[6]

4.3.3 NMT Setup

We build strong Transformer models on two high-resource language pairs: English→Chinese (En-Zh) and German→English (De-En), which tend to produce acceptable translation outputs on average, and thus hallucinations might be more misleading and harmful.

Data For En-Zh, we use the 18M training samples from WMT18 (Bojar et al., 2018) and *newsdev2017* as the validation set. For De-En, we use all training corpora from WMT21 (Wenzek et al., 2021) except for ParaCrawl, which yields 5M sentence pairs after cleaning as in Chen et al. (2021).[7] We use *newstest2019* for validation. We tokenize English and German sentences using the Moses scripts (Koehn et al., 2007) and Chinese sentences using the Jieba segmenter.[8] For En-Zh, we train separate BPE models for English and Chinese using 32k merging operations for each language. For De-En, we train a joint BPE model using 32k merging operations.

Models All models are based on the *base* Transformer (Vaswani et al., 2017). We apply label smoothing of 0.1. We train all models using the Adam optimizer (Kingma and Ba, 2015) with initial learning rate of 4.0 and batch sizes of 4,000 tokens for maximum 800k steps. We decode with beam search with a beam size of 4. The resulting NMT models achieve close or higher BLEU

[6] To compute the attention matrix, we average the attention weights over all attention heads in multi-head attention.

[7] https://github.com/browsermt/students/tree/master/train-student/clean

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

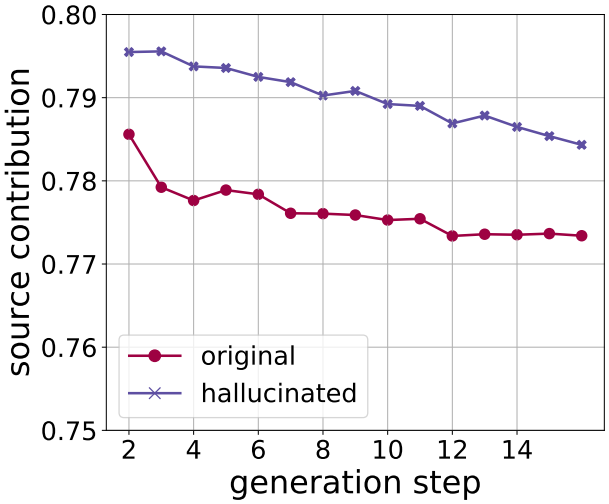


Figure 4.1: Relative source contributions $\sum_i R_t(x_i)$ at varying generation step t averaged over the original or hallucinated samples under a mixture of the misspelling, title-casing, and insertion perturbations.

scores than comparable published results.[9]

4.3.4 Findings

We test the aforementioned hypotheses on the perturbation-based counterfactual hallucination dataset constructed on English→Chinese.

First, we test the **Low Source Contribution Hypothesis** by computing the relative source contributions $\sum_{i=1}^{n} R_t(x_i)$ at each generation step t, where n is the length of each source sequence.[10] We plot the average contributions over a set of samples in Figure 4.1. It shows that hallucinations under source perturbations have only slightly higher source contributions ($\Delta \approx 0.1$) than the original samples. This departs from previous observations on pseudo-hallucinations (Voita et al., 2021), where the relative source contributions were lower on pseudo-hallucinations than on reference translations, perhaps because actual model outputs differ from pseudo-

[9]The En-Zh model achieves 33.5 BLEU on *newstest2017*, which is close to the 34.5 achieved by the most comparable model in Xu and Carpuat (2018). The De-En model achieves 35.0 BLEU on *newstest2019*, which is higher than the strong baseline (29.6 BLEU) from Germann (2020).

[10]Since LRP ensures that the sum of source and target contributions at each generation step is a constant, we only visualize the relative source contributions.

hallucinations created by inserting random target prefixes. We show that the hypothesis does not hold on actual hallucinations generated by the model itself.

To explain this phenomenon, we further examine the source contribution from the end-of-sequence (EOS) token, as previous works hypothesize that a translation is likely to be a hallucination when the attention distribution is concentrated on the source EOS token, which carries little information about the source (Berard et al., 2019b; Raunak et al., 2021). However, this hypothesis is only verified by qualitative analysis on individual samples. Our quantitative results on the perturbation-based hallucination dataset do not support this hypothesis and are more in line with the finding in Guerreiro et al. (2022) that the proportion of attention paid to the EOS token is not indicative of hallucinations. Specifically, our results show that the proportion of source contribution from the EOS token is slightly higher on the original samples (11.2%) than that on the hallucinated samples (10.8%). We will show in the next part that the source contribution is more concentrated on the beginning than the end of the source sentence when the model hallucinates.

Second, we test the **Local Source Contribution Hypothesis** by computing the **High-Contribution Ratio** $r(\lambda_0)$, which is the ratio of source tokens with normalized contribution $\bar{R}(x_i)$ larger than a threshold λ_0:

$$r(\lambda_0) = \sum_{i=1}^{n} \mathbb{I}(\bar{R}(x_i) > \lambda_0)/n \tag{4.7}$$

The ratio will be lower on the hallucinated samples than on the original ones if the hypothesis holds. We compute the standardized mean difference in High-Contribution Ratio between the hallucinated and original samples (Table 4.2).[11] The negative score differences in LRP-based

[11] We select the threshold λ_0 that leads to the largest score differences for each type of measurement.

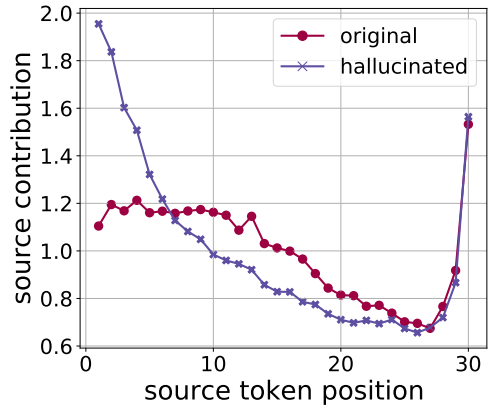
	Contrib Ratio		Staticity	
	D	N	D	N
Attention	-1.03†	+0.51†	3.39†	0.31†
LRP	**-1.05**†	**-1.13**†	**3.66**†	**2.74**†

Table 4.2: Standardized mean difference in High-Contribution Ratio (*Contrib Ratio*) and Source Contribution Staticity (*Staticity*) (computed on attention and LRP-based contribution matrices) between pairs of hallucinated and original samples. We show the score differences on degenerated (*D*) and non-degenerated (*N*) hallucinations separately. † indicates that the difference is statistically significant with $p < 0.05$.

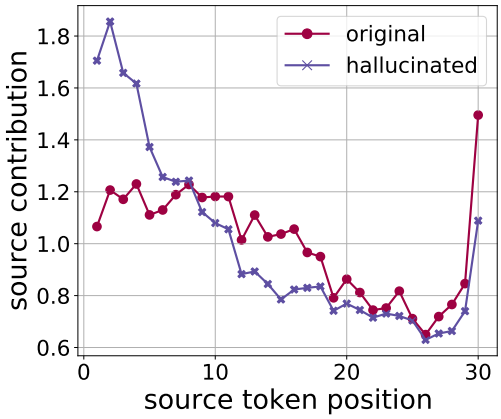
scores confirm the hypothesis, which is consistent with the findings of Lee et al. (2018b) based on attention weights. However, the pattern in attention-based scores is not consistent on degenerated and non-degenerated samples.

Furthermore, we investigate whether there is any positional bias for the local source contribution. We visualize the normalized source contribution $\bar{R}(x_i)$ averaged over all samples with a source length of 30 in Figure 4.2. The source contribution of the hallucinated samples is disproportionately high at the beginning of a source sequence. By contrast, on the original samples, the normalized contribution is higher at the end of the source sequence, which could be a way for the model to decide when to finish generation. The positional bias exists not only on hallucinations under insertions at the beginning of the source, but also on hallucinations under misspelling and title-casing perturbations that are applied at random positions.

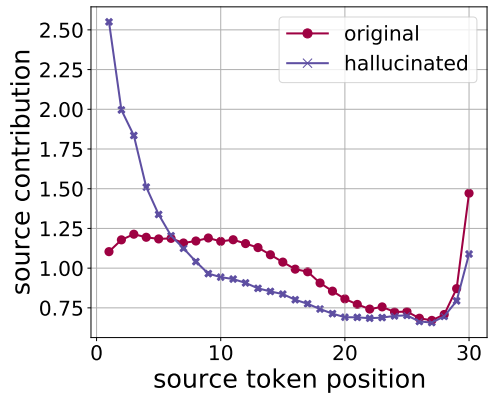
Third, we examine the **Static Source Contribution Hypothesis** hypothesis by first visualizing the source contributions $R_t(x_i)$ at varying source and generation positions on individual pairs of original and hallucinated samples. The heatmap of source contributions for the example from Table 4.1 are shown in Figure 4.3. On the original outputs, the source contribution distribution in each column changes dynamically when moving horizontally along target generation steps. By contrast, when the model hallucinates, the source contribution



(a) Misspelling



(b) Title-Casing

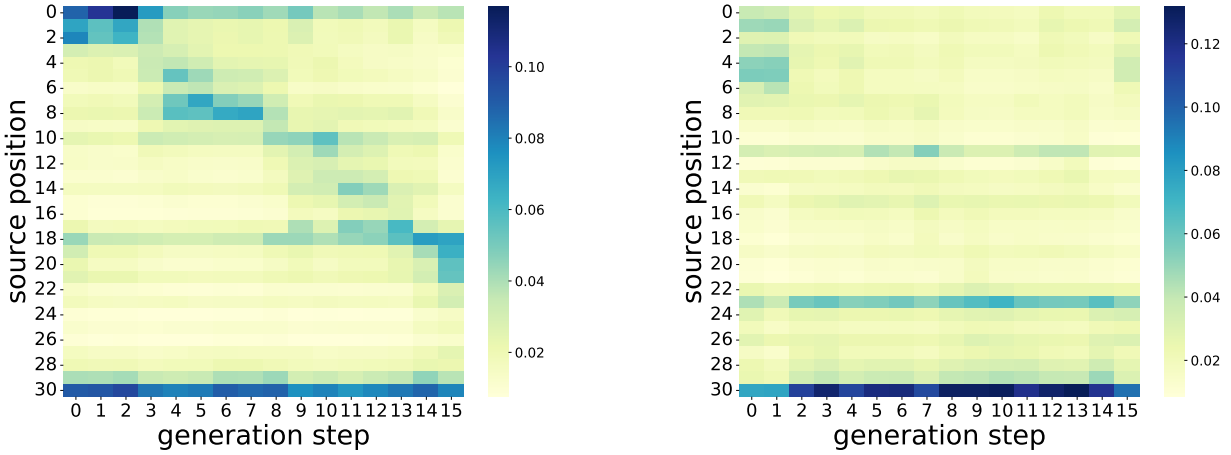


(c) Insertion

Figure 4.2: Normalized source contribution $\bar{R}(x_i)$ (Eq. 4.6) at each source token position averaged over the original or hallucinated samples under (a) misspelling, (b) title-casing, and (c) insertion perturbations.

distribution remains roughly static.

To quantify this pattern, we introduce **Source Contribution Staticity**, which measures how the source contribution distribution shifts over generation steps. Specifically, given a



(a) Original (b) Hallucinated

Figure 4.3: Heatmap of relative contributions of source tokens (y-axis) at each generation step (x-axis) computed on the example of the original translation and the counterfactual hallucination from the perturbed source in Table 4.1. The source contribution distribution remains static across almost all generation steps on the hallucinated sample, unlike in the original sample.

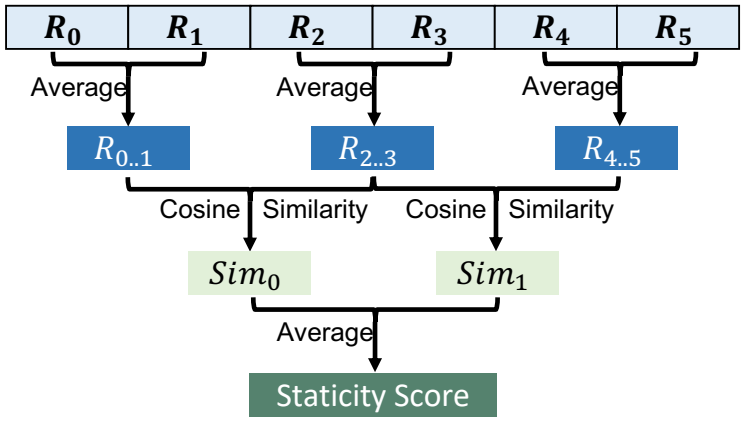


Figure 4.4: Computing the Source Contribution Staticity of window size $k = 2$ given the source contribution vectors $\boldsymbol{R_t} = [R_t(x_0)...R_t(x_n)]$ at generation step t.

window size k, we first divide the target sequence into several non-overlapping segments, each containing k tokens. Then, we compute the average vector over the contribution vectors $\boldsymbol{R_t} = [R_t(x_0)...R_t(x_n)]$ at step t within each segment. Finally, we measure the cosine similarity between the average contribution vectors of adjacent segments and average over the cosine similarity scores at all positions as the final score s_k of window size k. Figure 4.4 illustrates this process for a window size of 2.

Table 4.2 shows the standardized mean difference in Source Contribution Staticity between

the hallucinated and original samples in the degenerated and non-degenerated groups, taking the maximum staticity score among window sizes $k \in [1,3]$ for each sample. The positive differences in LRP-based scores supports the Static Source Contribution Hypothesis – the source contribution distribution is more static on the hallucinated samples than that on the original samples. Furthermore, LRP distinguishes hallucinations from non-hallucinations better than attention, especially on non-degenerated samples where the translation outputs contain no repetitive loops.

In summary, we find that, when generating a hallucination under source perturbations, the NMT model tends to rely on a small proportion of the source tokens, especially the tokens at the beginning of the source sentence. In addition, the distribution of the source contributions is more static on hallucinated translations than that on non-hallucinated translations. We turn to applying these insights on natural hallucinations next.

4.4 A Classifier to Detect Natural Hallucinations

Based on the above findings, we introduce a hallucination detector using features extracted from the distribution of source contributions, which is trained solely on perturbation-based samples that can be constructed automatically at scale.

Classifier We build a small multi-layer perceptron (MLP) with a single hidden layer and the following input features:

- **Normalized Source Contribution** of the first K_1 source tokens and the last K_1 source tokens: $\bar{R}(x_i)|i = 1,...,K_1, n-K_1,...,n-1$ (where n is the length of the source sequence and K_1 is a hyper-parameter), as we showed in the Local Source Contribution Hypothesis

that the contributions of the beginning and end tokens distribute differently between hallucinated and non-hallucinated samples.

- **Source Contribution Staticity** s_k given the source contributions $R_t(x_i)$ and a window size k as defined in Section 4.3.4. We include the similarity scores of window sizes $k = \{1, 2, ..., K_2\}$ as input features, where K_2 is a hyper-parameter.

This yields small classifiers with input dimension of 9 to 13. For each language pair, We train 20 classifiers with different random seeds and select the model with the highest validation F1 score.

Data Generation We construct the training and validation data using the same approach to constructing the perturbation-based hallucination dataset (Section 4.3.1), but with longer seed pairs – we randomly select seed sentence pairs with source length between 20 and 60 from the training corpora. We split the synthetic data randomly into the training (around $1k$ samples) and validation (around 200 samples) sets with roughly equal number of positive and negative samples.

4.5 Detecting Natural Hallucinations

To test how this hallucination classifier built upon insights from perturbation-based hallucinations generalizes to more realistic settings, we evaluate it on a human-annotated test bed for hallucinations generated on natural source inputs, and compare it against a wide range of relevant models.

4.5.1 Natural Hallucination Evaluation Set

To the best of our knowledge, the only publicly available dataset of annotated MT hallucinations is the Chinese-English test set by Zhou et al. (2021), which labels partial

	En-Zh	De-En
Detached hallucination	69	124
Non hallucination, including:		
Faithful translation	40	62
Incomplete translation	78	10
Locally unfaithful	26	13
Incomprehensible but aligned	3	21
Total	216	230

Table 4.3: Human annotation label distribution on the En-Zh and De-En natural hallucination test set (with random tie breaking on fine-grained labels; there are no ties on binary labels post-aggregation).

hallucinations at the token level. We build a test bed for detached hallucination detection for different language pairs and translation directions (En-Zh and De-En), and release the data together with the underlying NMT models (described in Section 4.3.3).

Since hallucinations are rare, we collect samples from large pools of out-of-domain data for our models to obtain enough positive examples of hallucinations for a meaningful test set. We use TED talk transcripts from the IWSLT15 training set (Cettolo et al., 2015) for En-Zh, and the JRC-Acquis corpus (Steinberger et al., 2006) of legislation from the European Union for De-En. We sample lower BLEU examples to increase the chances to find hallucinations, resulting in 216 and 230 samples for En-Zh and De-En respectively.[12]

Three bilingual annotators assess the faithfulness of the NMT output given each input. While we ultimately need a binary annotation of outputs as hallucinated or not, annotators were asked to choose one of five labels as it was found to improve annotation consistency in a pilot study:

- *Detached hallucination*: a translation with large segments that are unrelated to the source.
- *Faithful translation*: a translation that is faithful to the source.
- *Incomplete translation*: a translation that is partially correct but misses part(s) of the source.

[12] We select En-Zh samples with length > 30 and $BLEU < 0.1$; and De-En inputs with length > 20 and $BLEU < 0.3$.

- *Locally unfaithful*: a translation that contains a few unfaithful phrases but is otherwise faithful.
- *Incomprehensible but aligned*: a translation that is incomprehensible even though most phrases can be aligned to the source.

We then aggregate all labels except for the "detached hallucination" into the "non-hallucination" category. The inter-annotator agreement on aggregated labels is substantial for En-Zh, with a Fleiss's Kappa (Fleiss, 1971) score of $FK = 0.70$, and almost perfect for De-En, with $FK = 0.83$. This yields 32% of detached hallucinations in En-Zh and 54% on De-En. The non-hallucinated NMT outputs span all the fine-grained categories above, as can be seen in Table 4.3. Hallucinations are over-represented compared to what one might expect in the wild, but this is necessary to provide enough positive examples of hallucinations for evaluation.

4.5.2 Experimental Conditions

4.5.2.1 Introspection-based Classifiers

We implement the **LRP-based classifier** as described in Section 4.4. To lower the computational cost when computing the source contributions, we clip the source length at 40 and only consider the influence back-propagated through the most recent 10 target tokens, as prior work shows that nearby context is much more influential than long-range context (Khandelwal et al., 2018). We tune the hyper-parameters K_1 and K_2 within the space $K_1 \in \{1,3,5,7,9\}$, $K_2 \in \{4,8,12,16\}$ based on the average F1 accuracy on the validation set over three runs.

We compare it with an **attention-based classifier**, which uses the same features, but computes token contributions using attention weights averaged over all attention heads.

4.5.2.2 Model-free Baselines

We use three simple baselines to characterize the task. The **random classifier** that predicts hallucination with a probability of 0.5. The **degeneration** detector marks as hallucinations degenerated outputs that contain K more repetitive n-grams than the source, where K is a hyper-parameter tuned on the perturbation-based hallucination data. The **NMT probability scores** are used as a coarse model signal to detect hallucinations based on the heuristic that the model is less confident when producing a hallucination. The output is classified as a hallucination if the probability score is lower than a threshold tuned on the perturbation-based hallucination data.

4.5.2.3 Quality Estimation Classifier

We also compare the introspection-based classifiers with a baseline classifier based on the state-of-the-art quality estimation model – **COMET-QE** (Rei et al., 2020b). Given a source sentence and its NMT translation, we compute the COMET-QE score and classify the translation as a hallucination if the score is below a threshold tuned on the perturbation-based validation set.

4.5.2.4 Large Pre-trained Classifiers

We further compare the introspection-based classifiers with classifiers that rely on large pre-trained multilingual models, to compare the discriminative power of the source contribution patterns from the NMT model itself to extrinsic semantically-driven discrimination criteria.

We use the cosine distance between the **LASER** representations (Artetxe and Schwenk, 2019; Heffernan et al., 2022) of the source and the NMT translation. It classifies a translation as a hallucination if the distance score is higher than a threshold tuned on the perturbation-based

validation set.

Inspired by (Zhou et al., 2021), we build an **XLM-R classifier** by fine-tuning the XLM-R model (Conneau et al., 2020) on synthetic hallucination samples. We randomly select $50K$ seed pairs of source and reference sentences with source lengths between 20 and 60 from the parallel corpus and use the following perturbations to construct examples of detached hallucinations:

- Map a source sentence to a random target from the parallel corpus to simulate natural, detached hallucinations.
- Repeat a random dependency subtree in the reference many times to simulate degenerated hallucinations.
- Drop a random clause from the source sentence to simulate natural, detached hallucinations.

We then collect diverse non-hallucinated samples:

- Original seed pairs provide faithful translations.
- Create incomplete translations by randomly dropping a dependency subtree from references.
- Randomly substitute a phrase in the reference keeping the same part-of-speech to simulate translations with locally unfaithful phrases.

The final training and validation sets contain around $300k$ and 700 samples, respectively. We fine-tune the pre-trained model with a batch size of 32. We use the Adam optimizer (Kingma and Ba, 2015) with decoupled weight decay (Loshchilov and Hutter, 2019) and an initial learning rate of 2×10^{-5}. We fine-tune all models for 5 epochs and select the checkpoint with the highest F1 score on the validation set.

	Params	De-En				En-Zh			
		P	R	F1	AUC	P	R	F1	AUC
Model-free Baselines									
Random	0	51.4	49.5	50.4	46.7	34.0	52.4	41.2	51.4
Degeneration	1	59.3	56.5	57.9	–	83.1	69.0	75.4	–
NMT Score	1	66.7	9.7	16.9	47.7	32.6	**98.6**	49.0	74.2
Quality Estimation Classifier									
COMET-QE	363*M*	77.7	75.8	76.7	81.3	33.7	**100.0**	50.4	94.5
Large Pre-trained Classifiers									
LASER	45*M*	**98.5**	54.0	69.8	**97.1**	88.5	77.1	82.4	**97.7**
XLM-R	125*M*	**96.6**	21.8	35.3	49.1	**95.2**	**93.3**	**94.2**	**97.2**
Introspection-based Classifiers									
Attention-based	< 300	53.9	67.2	59.7	57.7	59.0	91.4	71.5	90.0
LRP-based	< 300	86.1	**94.3**	**90.0**	96.1	**98.5**	**93.9**	**96.2**	**99.8**
Ensemble Classifier									
LRP + LASER	45*M*	**98.5**	52.0	68.1	–	**99.4**	74.6	85.3	–

Table 4.4: Precision (P), Recall (R), F1 and Area Under the Receiver Operating Characteristic Curve (AUC) scores of each classifier on English-Chinese (En-Zh) and German-English (De-En) NMT outputs (means of three runs). We boldface the highest scores based on independent student's t-test with Bonferroni Correction ($p < 0.05$). The *Params* column indicates the total number of parameters used for each method (in addition to the NMT parameters).

4.5.3 Findings

As shown in Table 4.4, we compare different classifiers against the baselines by the Precision, Recall and F1 scores. Since false positives and false negatives might have a different impact in practice (e.g., does the detector flag examples for review by humans, or entirely automatically? what is MT used for?), we also report the Area Under the Receiver Operating Characteristic Curve (AUC), which characterizes the discriminative power of each method at varying threshold settings.

Main Results The LRP-based and the LASER classifiers are the best hallucination detectors, reaching AUC scores in the high 90s for both language pairs, which is considered outstanding

discrimination ability (Hosmer Jr et al., 2013).

The LRP-based classifier is the best hallucination detector overall, with slightly lower AUC than LASER on De-En, but more balanced Precision and Recall leading to F1 scores of 90.0 on De-En and 96.2 on En-Zh, improving over LASER by 20 and 14 points respectively. This shows that the source contribution patterns identified on hallucinations under perturbations (Section 4.3) generalize as symptoms of natural hallucinations, although the surface-form patterns in the perturbed source and the natural source that leads to hallucinations are very different.[13] It also confirms that LRP provides a better signal to characterize token contributions than attention, improving F1 by 25-30 points and AUC by 10-38 points. These high scores represent large improvements of 26-49 points on AUC and 21-73 points on F1 over the model-free baselines.

Model-free Baselines The model-free baselines shed light on the nature of the hallucinations found in the dataset. The degeneration baseline is the best among them, with 57.9 F1 on De-En and 75.4 F1 on En-Zh, suggesting that the Chinese hallucinations are more frequently degenerated than the English hallucinations from German. However, ignoring the remaining hallucinations is problematic, since they might be more fluent and thus more likely to mislead readers. The NMT score is a poor predictor which scores often worse than the random baseline, in line with the previous finding that NMT scores do not capture faithfulness well during inference (Wang et al., 2020). Manual inspection shows that the NMT score can be low when the output is faithful to the source but contains infrequent tokens. On the other hand, the NMT score of a hallucinated output can be high when it contains mostly frequent tokens.

[13] Based on our inspection, the natural inputs that lead to hallucinations do not necessarily contain the specific types of perturbations that trigger hallucinations in the synthetic case.

Quality Estimation Classifier The COMET-QE classifier achieves higher AUC and F1 scores than the model-free classifiers, except for En-Zh, where the degeneration baseline obtains higher F1 than the COMET-QE classifier. However, compared with the LASER classifier which also relies on a large pre-trained encoder, COMET-QE lags behind on AUC scores and obtains much lower F1 on En-Zh with only slightly higher F1 on De-En. It also underperforms the LRP-based classifier on both AUC and F1. This is consistent with previous findings that quality estimation models trained on data with insufficient negative samples (e.g. COMET-QE) are inadequate for detecting critical MT errors such as hallucinations (Guerreiro et al., 2022; Sudoh et al., 2021; Takahashi et al., 2021).

Pre-trained Classifiers These classifiers achieve high precision, approaching that of the LRP-based model on En-Zh and surpassing it on De-En, but they lag behind in Recall, particularly on De-En, where their Recall is close or worse to that of the random baseline. This suggests that they suffer from domain shift, which is bigger in De-En (News→Law) than En-Zh (News→TED). By contrast the introspection-based classifiers are more robust. LASER achieves higher AUC than the XLM-R classifier, approaching that of the LRP-based one on En-Zh and surpassing it on De-En. This suggests that, although the LASER threshold tuned on the perturbation-based hallucination data generalizes poorly to natural hallucinations, LASER is still a good scoring model for identifying hallucinations. Furthermore, the Recall of the LASER classifier does not degrade as dramatically from En-Zh to De-En as that of the XLM-R classifier, which needs to be fine-tuned on large amounts of synthetic training data and generalizes more poorly across domains.

	De-En		En-Zh	
	F1	AUC	F1	AUC
All features	**90.0**	**96.1**	**96.2**	**99.8**
- Src Contrib	76.4	**96.1**	91.4	99.2
- Staticity	76.5	80.1	72.9	87.5

Table 4.5: Ablating the Normalized Source Contribution (*Src Contrib*) and Source Contribution Staticity (*Staticity*) features used in the LRP-based classifier. We boldface the highest scores based on independent student's t-test with Bonferroni Correction ($p < 0.05$).

Source: C) DASS DIE WAREN IN DEM ZUSTAND IN DIE GEMEINSCHAFT VERSANDT WORDEN SIND, IN DEM SIE ZUR AUSSTELLUNG GESANDT WURDEN;
Correct Translation: C) THAT THE GOODS WERE SHIPPED TO THE COMMUNITY IN THE CONDITION IN WHICH THEY ARE SENT FOR EXHIBITION;
Output: C) THAT THE WOULD BE CONSIDERED IN THE COMMUNITY, IN which YOU WILL BE EXCLUSIVE;

Table 4.6: Example of a detached hallucination produced by the De-En NMT being classified as non-hallucination by the LRP-based classifier.

LRP + LASER The LRP-based and LASER classifiers emerge as the top-2 classifiers, but they make different errors – the confusion matrix comparing their predictions shows that the two classifiers agree on 70% and 91% of samples on De-En and En-Zh, respectively. Thus an ensemble that detects hallucinations when the LRP and LASER classifiers both do so, yields a very high precision classifier (at the expense of recall).

LRP Ablations The LRP-based classifier benefits the most from Source Contribution Staticity features (Table 4.5). Removing them hurts AUC by 12-16 points and F1 by 14-23, confirming that the Static Source Contribution Hypothesis holds on natural hallucinations. Ablating the Normalized Source Contribution features also causes a significant drop in both F1 and AUC on En-Zh, and hurts F1 significantly but not AUC on De-En.

Error Analysis Incomprehensible but aligned translations suffer from the highest false positive rate for the LRP classifier, followed by incomplete translations. Additionally, the classifier can fail to detect hallucinations caused by the mistranslation of a large span of the source with rare or previously unseen tokens, rather than by pathological behavior at inference time as shown by the example in Table 4.6.

Toward Practical Detectors Detecting hallucinations in the wild is challenging since they are often extremely rare and the fraction of hallucinations may vary greatly depending on specific test cases. We provide a first step in this direction by stress testing the top classifiers in an in-domain scenario where hallucinations are expected to be rare. Specifically, we randomly select 10k English sentences from the *"News Crawl: articles from 2021"* from WMT21 (Wenzek et al., 2021) and use the En-Zh NMT model to translate them into Chinese. We measure the *Precision@20* for hallucination detection by manually examining the top-20 highest scoring hallucination predictions for each method. The LASER and LRP-based classifiers evaluated above (without fine-tuning in this setting) achieve 35% and 40% Precision@20 (compared to 0% for the random baseline). More interestingly, after tuning threshold on the predicted probabilities (which is originally set to 0.5) so that each classifier predicts hallucination 1% of the time, the LRP + LASER ensemble detects 13 hallucinations, with a much higher precision of 85%. LRP + LASER thus have the potential to provide useful signals for detecting hallucinations even when they are needles in a haystack.

4.6 Connecting Hallucination Symptoms to Spurious Biases

Having discovered the symptoms that are indicative of hallucinations at inference time, we further investigate how these symptoms connect with the spurious biases learned at training time. To this end, we first extract the training samples that contain similar target segments to one of the hallucination outputs in the natural hallucination evaluation set. We then examine the source contribution patterns on the extracted training samples and compare them with the patterns on the hallucination samples at inference time.

4.6.1 Extracting Hallucination-Related Training Samples

We consider a training sample as hallucination-related if it contains similar target segments to a hallucinated output. Specifically, for each hallucination output in the De-En natural hallucination evaluation set, we extract the top-10 most similar training samples based on the BLEU score between the hallucination output and the reference translation of each sample.[14] Next, we rank the extracted samples based on the BLEU similarity and select the top-200 samples as hallucination-related for degenerated and non-degenerated hallucinations (divided based on the same criterion in Section 4.3.1) separately. Finally, to construct the contrastive group, we randomly select 200 samples from the training corpora.

[14] We conduct the experiment on the De-En evaluation set because it contains a more balanced number of degenerated versus non-degenerated hallucinations.

	Contrib Ratio		Staticity	
	D	N	D	N
LRP	-0.96†	-0.90†	1.38†	1.16†

Table 4.7: Standardized mean difference in High-Contribution Ratio (*Contrib Ratio*) and Source Contribution Staticity (*Staticity*) computed on LRP-based contribution matrices between hallucination-related and contrastive samples. We show the score differences on samples related to degenerated (*D*) and non-degenerated (*N*) hallucinations separately. † indicates that the difference is statistically significant with $p < 0.05$.

4.6.2 Analyzing Relative Token Contributions on Training Samples

We hypothesize that the source contribution patterns discovered on hallucination samples are connected with the spurious biases learned at training time. To test this hypothesis, we measure the High-Contribution Ratio and Source Contribution Staticity on the training samples related to degenerated or non-degenerated hallucinations and compare the scores with those on the contrastive samples. As shown in Table 4.7, the hallucination-related samples have significantly lower High-Contribution Ratio and higher Source Contribution Staticity than the contrastive samples, which indicates the spurious bias that the prediction of the ground-truth tokens on hallucination-related samples are based statically on a small subset of the source tokens. The trend is similar to Table 4.2 where we compare the High-Contribution Ratio and Source Contribution Staticity on hallucination versus non-hallucination samples. Thus, it verifies our hypothesis that the hallucination symptoms are related to the spurious biases learned during training. This connects with the previous finding that hallucinations under source perturbations are more likely to occur on the long-tail samples memorized by the NMT model (Raunak et al., 2021). However, prior work did not show whether the model tends to memorize the source or target side of the samples or how this finding generalize to natural hallucinations. This work is complementary to the previous work by showing that hallucinations, either on perturbed or

natural source inputs, can be explained by the target memorization issue on certain training samples where the model tends to memorize the target sequence while ignoring a large part of the source sentence.

4.7 Limitations

The findings of this work should be interpreted with several limitations in mind. First, we focus on understanding and detecting detached hallucinations in MT. As a result, our findings do not elucidate the internal model symptoms that lead to partial hallucinations (Zhou et al., 2021), although the methodology in this work could be used to shed light on this question. Second, our experiments are based on the NMT models trained using the parallel data from WMT without exploiting monolingual data or comparable corpora retrieved from collections of monolingual texts (e.g. WikiMatrix (Schwenk et al., 2021)). Therefore, it remains an open question how the discovered hallucination symptoms generalize to NMT models trained on more diverse data types. Finally, we primarily test the hallucination classifiers in the settings where the number of hallucinations and non-hallucinations are roughly balanced, while in practice, hallucinations are expected to be rare. To test how the classifiers perform in this setting, we conduct a small-scale experiment on an in-domain dataset and show promising results using our LRP +LASER ensemble classifier. However, further experiments are needed to systematically evaluate how these classifiers can be used for hallucination detection in the wild.

4.8 Summary

We contributed a thorough empirical study of the notorious but poorly understood hallucination phenomenon in NMT, which shows that internal model symptoms exhibited during inference are strong indicators of hallucinations. Using counterfactual hallucinations triggered by perturbations, we showed that distinctive source contribution patterns alone indicate hallucinations better than the relative contribution of the source and target. We further show that our findings can be used for detecting natural hallucinations much more accurately than model-free baselines and quality estimation models. Our detector also outperforms black-box classifiers based on pre-trained models. We also release human-annotated test beds of natural English-Chinese and German-English hallucinations to enable further research. Finally, we connected the internal symptoms of hallucinations to specific spurious biases learned at training time. This work provides a path toward mitigating hallucinations by designing better supervised learning algorithms to combat these spurious biases. In the next chapter, we turn to study semi-supervised learning algorithms with stronger inductive biases to improve the sample-efficiency of NMT models.

Chapter 5: Stronger Inductive Biases for Integrating Language Priors

5.1 Introduction

Besides supervised learning algorithms that mitigate undesirable biases, we can also build more sample-efficient NMT models through semi-supervised learning algorithms to better exploit the unsupervised monolingual data. In low-resource settings where the amount of supervised (parallel) data is limited, effective learning of the prior language distributions from the more abundant monolingual data can substantially reduce the amount of supervised data needed for NMT training. In this chapter, we study the inductive biases in existing semi-supervised learning algorithms by introducing a theoretical framework with a provable guarantee on its global optimum to unify and compare existing algorithms.

Existing semi-supervised learning algorithms (including Back-Translation (Sennrich et al., 2016b), Iterative Back-Translation (Cotterell and Kreutzer, 2018; Zhang et al., 2018), and Dual Learning (He et al., 2016)) primarily rely on unrelated heuristic optimization objectives, and it is not clear what their respective strengths and weaknesses are, nor how they relate to the ideal but intractable objective of maximizing the marginal likelihood of the monolingual data (i.e., $p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p_\theta(\boldsymbol{y}|\boldsymbol{x})q(\boldsymbol{x})$ given target sentences $\boldsymbol{y}$, an NMT model $p_\theta(\boldsymbol{y}|\boldsymbol{x})$, and the prior distribution $q(\boldsymbol{x})$ on source $\boldsymbol{x}$).

Instead of proposing new empirical methods, this work introduces a theoretical framework,

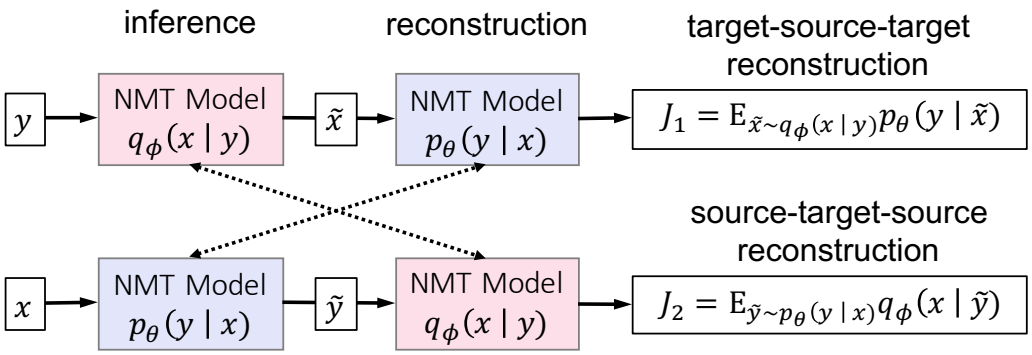


Figure 5.1: Our dual reconstruction objective sums 1) a target-source-target objective $\mathscr{J}_1$ on target sentences $\boldsymbol{y}$ using the NMT model $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ for inference and $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ for reconstruction, and 2) a source-target-source objective $\mathscr{J}_2$ on source sentences $\boldsymbol{x}$ using $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ for inference and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ for reconstruction. Models connected by dotted arrows share parameters.

namely the **dual reconstruction objective**, that unifies and explains these established techniques, sheds new light on why they work and how they compare (Figure 5.1). In Section 5.2, we show that, under some assumptions, this objective remarkably shares the same global optimum as the intractable marginal likelihood objective where the model's marginal distribution $p_\theta(\boldsymbol{y})$ coincides with the target sentence distribution $p(\boldsymbol{y})$. We also show that Iterative Back-Translation (IBT) and Dual Learning can be viewed as different ways to approximate its optimization.

Theory suggests that IBT approximates the dual reconstruction objective more closely than the more complex Dual Learning approach, and in particular that Dual Learning's additional language model loss is redundant. We investigate whether these differences matter in practice by conducting the first controlled empirical comparison of Back-Translation, IBT, and Dual Learning in high-resource (WMT de-en), low-resource (WMT tr-en), and cross-domain settings (News→TED, de-en). Results support our theory that the additional language model loss and policy gradient estimation in Dual Learning is redundant and show that IBT outperforms the more complex Dual Learning algorithm in terms of translation quality. Furthermore, we also compare different optimization strategies used in IBT to better balance translation quality against the computational cost.

5.2 Theoretical View with Dual Reconstruction Objective

5.2.1 Variational Auto-Encoders for Semi-Supervised MT

Following Cotterell and Kreutzer (2018), we define a generative latent variable model of bitext

$$p_\theta(\boldsymbol{x},\boldsymbol{y}) = p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x})q(\boldsymbol{x})$$

where the source $\boldsymbol{x}$ is randomly sampled from the prior distribution $q(\boldsymbol{x})$ estimated by the empirical data distribution $q_{data}(\boldsymbol{x})$ based on the abundant source monolingual data $\mathscr{M}_X = \{\boldsymbol{x}^{(m)}\}_{m=1}^{M}$:

$$q_{data}(\boldsymbol{x}) = \begin{cases} \frac{1}{|\mathscr{M}_X|}, & \text{if } \boldsymbol{x} \in \mathscr{M}_X \\ 0, & \text{otherwise} \end{cases}$$

and the target translation $\boldsymbol{y}$ is sampled from the translation model $p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x})$ conditioned on $\boldsymbol{x}$.

Given the target sentence distribution $p(\boldsymbol{y})$ estimated by the empirical data distribution $p_{data}(\boldsymbol{y})$ of target monolingual data $\mathscr{M}_Y = \{\boldsymbol{y}^{(m)}\}_{m=1}^{M'}$, we can view $\boldsymbol{x}$ as a latent variable and maximize the marginal log-likelihood

$$\mathscr{J}_u(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y})\right]$$

where $p_\theta(\boldsymbol{y})$ is the model's marginal likelihood $p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p_\theta(\boldsymbol{x},\boldsymbol{y})$. The global optimum of the objective is achieved when the model's marginal distribution $p_\theta(\boldsymbol{y})$ perfectly matches the target sentence distribution $p(\boldsymbol{y})$.[1]

[1]We will define constraints to guarantee avoiding the uninteresting solution where $p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x}) = p(\boldsymbol{y})$ in

However, directly optimizing the marginal likelihood $p_\theta(\boldsymbol{y})$ is intractable due to the infinite space of $\boldsymbol{x}$. We can instead apply variational auto-encoding (VAE) models by introducing an inference network $p_\psi(\boldsymbol{x}|\boldsymbol{y})$ and maximize the variational lower-bound (ELBO) of $\log p_\theta(\boldsymbol{y})$:

$$\begin{aligned}\log p_\theta(\boldsymbol{y}) \geq & \mathbb{E}_{\boldsymbol{x}\sim p_\psi(\boldsymbol{x}|\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y}|\boldsymbol{x})\right] \\ & - D_{\mathrm{KL}}\left[\, p_\psi(\boldsymbol{x}|\boldsymbol{y}) \middle|\middle| q(\boldsymbol{x})\right]\end{aligned} \tag{5.1}$$

where $D_{\mathrm{KL}}\left[\, p_\psi \middle|\middle| q\right]$ is the Kullback-Leibler (KL) divergence. However, estimating the prior distribution $q(\boldsymbol{x})$ by the discrete data distribution $q_{data}(\boldsymbol{x})$ makes it difficult to directly compute the KL term. One can estimate $q(\boldsymbol{x})$ using a language model (LM) trained to maximize the likelihood of the source monolingual data (Baziotis et al., 2019; Miao and Blunsom, 2016), at the cost of introducing additional model bias into the translation model. The non-differentiable KL term requires gradient estimators such as policy gradient (Williams, 1992) or Gumbel-softmax (Jang et al., 2017), which may introduce further training noise (He et al., 2020).

To address these issues, we introduce the dual reconstruction objective, which includes two reconstruction terms that resemble the first term in the ELBO objective (Eq. (5.1)) while excluding the KL term that is challenging to optimize and show that this objective has desirable properties and can be better approximated in practice.

Definition 5.1. Given prior distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ over the sentences $\boldsymbol{x}$ in the source language space Σ_x and $\boldsymbol{y}$ in the target language space Σ_y, we define the **dual reconstruction objective** $\mathscr{J}_{dual}(\theta,\phi)$ for dual translation models $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ as the sum of the target-

Section 5.2.2.

source-target objective $\mathscr{J}_1$ and source-target-source objective $\mathscr{J}_2$:

$$\begin{aligned}
\mathscr{J}_{dual}(\theta,\phi) &= \mathscr{J}_1(\theta,\phi) + \mathscr{J}_2(\theta,\phi) \\
\mathscr{J}_1(\theta,\phi) &= \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y}|\boldsymbol{x})\right]\right] \\
\mathscr{J}_2(\theta,\phi) &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\mathbb{E}_{\boldsymbol{y}\sim p_\theta(\boldsymbol{y}|\boldsymbol{x})}\left[\log q_\phi(\boldsymbol{x}|\boldsymbol{y})\right]\right]
\end{aligned} \tag{5.2}$$

For $\mathscr{J}_1$, the target-to-source model $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ serves as the **inference model** to produce pseudo source sequences $\tilde{\boldsymbol{x}}$ given target sequences $\boldsymbol{y}$ and $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ serves as the **reconstruction model** to reconstruct $\boldsymbol{y}$ given $\tilde{\boldsymbol{x}}$, and vice versa for $\mathscr{J}_2$. We first define the mutual information constraint in Section 5.2.2 and show in Section 5.2.3 that $\mathscr{J}_{dual}(\theta,\phi)$ shares the same global optimum as the marginal likelihood objective which is intractable to optimize directly.[2] In Section 5.2.4, we compare and contrast how IBT and Dual Learning approximate $\mathscr{J}_{dual}(\theta,\phi)$.

5.2.2 Mutual Information Constraint

The global optimum of the marginal likelihood objective is achieved when the model's marginal distribution $p_\theta(\boldsymbol{y}) = p(\boldsymbol{y})$. Given a translation model with enough capacity without any constraint on how the model output is dependent on the source context, this could lead to a degenerate solution $p_\theta(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y})$ where the model ignores the source input and memorizes the monolingual training data. We constrain the translation model to avoid this situation, using the mutual information of a conditional distribution $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ which measures how much $\boldsymbol{y}$ is dependent on $\boldsymbol{x}$ in p_θ (Hoffman and Johnson, 2016). Here, this mutual information measures the degree to which model translations depend on the source.

[2] We focus on key components of the proof and leave detailed derivations for supplemental material.

Definition 5.2. Given a prior distribution $q(\boldsymbol{x})$ over $\boldsymbol{x} \in \Sigma_x$, we define the **mutual information** I_{p_θ} of $\boldsymbol{x}$ and $\boldsymbol{y}$ in the conditional distribution $p_\theta(\boldsymbol{y}|\boldsymbol{x})$:

$$I_{p_\theta} = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left[\, p_\theta(\boldsymbol{y}|\boldsymbol{x}) || \, p_\theta(\boldsymbol{y})\right]\right] \tag{5.3}$$

where $p_\theta(\boldsymbol{y})$ is the marginal distribution:

$$p_\theta(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p_\theta(\boldsymbol{y}|\boldsymbol{x}) q(\boldsymbol{x}) \tag{5.4}$$

To avoid the degenerate solution, we constrain the model's mutual information by:

$$0 \leq I_{min} \leq I_{p_\theta} \leq I_{max} \leq \max_{p \in P_{XY}} I_p(\boldsymbol{x};\boldsymbol{y})$$

where I_{min} and I_{max} are pre-defined constant values between zero and the maximum mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ given any joint distribution $p(\boldsymbol{x},\boldsymbol{y}) \in P_{XY}$ whose marginals satisfy $\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y})$ and $\sum_{\boldsymbol{y}} p(\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{x})$. Hledík et al. (2019) prove that the maximum mutual information $\max_{p\in P_{XY}} I_p(\boldsymbol{x};\boldsymbol{y}) = \min(\mathbb{H}\left[q(\boldsymbol{x})\right], \mathbb{H}\left[p(\boldsymbol{y})\right])$, where $\mathbb{H}\left[q(\boldsymbol{x})\right]$ and $\mathbb{H}\left[p(\boldsymbol{y})\right]$ are the entropy of prior distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$. Thus, the maximum mutual information should be large enough to properly bound the model's mutual information if $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are defined on large monolingual corpora $\mathscr{M}_X$ and $\mathscr{M}_Y$.

Intuitively, the constraint requires that the model's mutual information cannot be so small that the model ignores the source context nor so large such that is not robust to the noise in the

source input. We will show in Section 5.3.4 that in practice, this constraint is met when jointly optimizing the supervised and unsupervised objectives without explicitly applying constrained optimization.

5.2.3 Understanding the Global Optimum of the Dual Reconstruction Objective

We first characterize the upper bound of the dual reconstruction objective.

Proposition 1. Given prior distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ over $\boldsymbol{x} \in \Sigma_x$ and $\boldsymbol{y} \in \Sigma_y$, if parameterized probability models p_θ and q_ϕ have enough capacity under the constraint that:

$$0 \leq I_{min} \leq I_{p_\theta}, I_{q_\phi} \leq I_{max} \leq \max_{p \in P_{XY}} I_p(\boldsymbol{x}; \boldsymbol{y})$$

where I_{min} and I_{max} are pre-defined constant values between zero and the maximum mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ given any joint distribution $p(\boldsymbol{x}, \boldsymbol{y}) \in P_{XY}$ whose marginals satisfy $\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y})$ and $\sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x})$. Then, the dual reconstruction objective is upper-bounded by $\mathscr{J}_{dual}(\theta, \phi) \leq 2I_{max} - \mathbb{H}[q(\boldsymbol{x})] - \mathbb{H}[p(\boldsymbol{y})]$, and the upper bound is achieved iff

$$\begin{aligned} I_{q_\phi} &= I_{max} \\ I_{p_\theta} &= I_{max} \\ p_\theta(\boldsymbol{y} \mid \boldsymbol{x}) &= \frac{q_\phi(\boldsymbol{x} \mid \boldsymbol{y})}{q_\phi(\boldsymbol{x})} p(\boldsymbol{y}) \\ q_\phi(\boldsymbol{x} \mid \boldsymbol{y}) &= \frac{p_\theta(\boldsymbol{y} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{y})} q(\boldsymbol{x}) \end{aligned} \tag{5.5}$$

Proof. First we prove that $\mathscr{J}_1(\theta,\phi) \leq I_{max} - \mathbb{H}[p(\boldsymbol{y})]$, and the upper bound is achieved iff

$$
\begin{aligned}
I_{q_\phi} &= I_{max} \\
p_\theta(\boldsymbol{y}|\boldsymbol{x}) &= \frac{q_\phi(\boldsymbol{x}|\boldsymbol{y})}{q_\phi(\boldsymbol{x})} p(\boldsymbol{y})
\end{aligned}
$$

where $\mathbb{H}[p(\boldsymbol{y})]$ is the entropy of the prior distribution $p(\boldsymbol{y})$.

To show this, we denote the posterior distribution $Q(\boldsymbol{y}|\boldsymbol{x}) = \frac{q_\phi(\boldsymbol{x}|\boldsymbol{y})}{q_\phi(\boldsymbol{x})} p(\boldsymbol{y})$, and rewrite $\mathscr{J}_1$:

$$
\begin{aligned}
\mathscr{J}_1 =& \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log p_\theta(\boldsymbol{y}|\boldsymbol{x})\right]\right] \\
=& \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log Q(\boldsymbol{y}|\boldsymbol{x})\right]\right] \\
&+ \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[log\frac{p_\theta(\boldsymbol{y}|\boldsymbol{x})}{Q(\boldsymbol{y}|\boldsymbol{x})}\right]\right] \\
=& \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log Q(\boldsymbol{y}|\boldsymbol{x})\right]\right] \\
&+ \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[log\frac{p_\theta(\boldsymbol{y}|\boldsymbol{x})q_\phi(\boldsymbol{x})}{q_\phi(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})}\right]\right] \\
=& \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log Q(\boldsymbol{y}|\boldsymbol{x})\right]\right] \\
&- D_{\mathrm{KL}}\left[\, q_\phi(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) \,\middle\|\, p_\theta(\boldsymbol{y}|\boldsymbol{x})q_\phi(\boldsymbol{x})\right] \\
=& \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log \frac{q_\phi(\boldsymbol{x}|\boldsymbol{y})}{q_\phi(\boldsymbol{x})}\right]\right] \\
&+ \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{y})}\left[\log p(\boldsymbol{y})\right]\right] \\
&- D_{\mathrm{KL}}\left[\, q_\phi(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) \,\middle\|\, p_\theta(\boldsymbol{y}|\boldsymbol{x})q_\phi(\boldsymbol{x})\right] \\
=& I_{q_\phi} - \mathbb{H}[p(\boldsymbol{y})] \\
&- D_{\mathrm{KL}}\left[\, q_\phi(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) \,\middle\|\, p_\theta(\boldsymbol{y}|\boldsymbol{x})q_\phi(\boldsymbol{x})\right]
\end{aligned}
$$

Since the KL divergence between two distributions is always non-negative and is zero iff they are

equal, we have

$$\mathscr{J}_1(\theta,\phi) \leq I_{q_\phi} - \mathbb{H}\left[p(\boldsymbol{y})\right] \leq I_{max} - \mathbb{H}\left[p(\boldsymbol{y})\right]$$

and $\mathscr{J}_1(\theta,\phi) = I_{max} - \mathbb{H}\left[p(\boldsymbol{y})\right]$ iff

$$I_{q_\phi} = I_{max}$$
$$D_{\mathrm{KL}}\left[\, q_\phi(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) \middle|\middle| p_\theta(\boldsymbol{y}|\boldsymbol{x})q_\phi(\boldsymbol{x})\right] = 0$$

The second equality holds iff

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{q_\phi(\boldsymbol{x}|\boldsymbol{y})}{q_\phi(\boldsymbol{x})}p(\boldsymbol{y})$$

Similarly, we can prove that $\mathscr{J}_2(\theta,\phi) \leq I_{max} - \mathbb{H}\left[q(\boldsymbol{x})\right]$, and the upper bound is achieved iff

$$I_{p_\theta} = I_{max}$$
$$q_\phi(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_\theta(\boldsymbol{y}|\boldsymbol{x})}{p_\theta(\boldsymbol{y})}q(\boldsymbol{x})$$

thus $\mathscr{J}_{dual}(\theta,\phi) \leq 2I_{max} - \mathbb{H}\left[q(\boldsymbol{x})\right] - \mathbb{H}\left[p(\boldsymbol{y})\right]$ and the upper bound is achieved iff θ and ϕ satisfy Eq. (5.5), concluding the proof. □

Proposition 1 shows that $\mathscr{J}_{dual}(\theta,\phi)$ has an upper bound that could be reached when the mutual information of $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ are maximized, and $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ are equal to the posterior distribution for each other. Next we show that the upper bound is indeed the global maximum of the objective $\mathscr{J}_{dual}(\theta,\phi)$, as there exists a solution for the above conditions.

Proposition 2. Given distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ over $\boldsymbol{x} \in \Sigma_x$ and $\boldsymbol{y} \in \Sigma_y$, if parameterized probability models p_θ and q_ϕ have enough capacity under the constraint that:

$$0 \leq I_{min} \leq I_{p_\theta}, I_{q_\phi} \leq I_{max} \leq \max_{p \in P_{XY}} I_p(\boldsymbol{x}; \boldsymbol{y}) \tag{5.6}$$

where I_{min} and I_{max} are pre-defined constant values between zero and the maximum mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ given any joint distribution $p(\boldsymbol{x}, \boldsymbol{y}) \in P_{XY}$ whose marginals satisfy $\sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y})$ and $\sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x})$. Then there exist θ^* and ϕ^* such that:

$$\begin{aligned} I_{q_{\phi^*}} &= I_{p_{\theta^*}} = I_{max} \\ p_{\theta^*}(\boldsymbol{y} \mid \boldsymbol{x}) &= \frac{q_{\phi^*}(\boldsymbol{x} \mid \boldsymbol{y})}{q_{\phi^*}(\boldsymbol{x})} p(\boldsymbol{y}) \\ q_{\phi^*}(\boldsymbol{x} \mid \boldsymbol{y}) &= \frac{p_{\theta^*}(\boldsymbol{y} \mid \boldsymbol{x})}{p_{\theta^*}(\boldsymbol{y})} q(\boldsymbol{x}) \end{aligned} \tag{5.7}$$

Proof. Since I_{max} satisfies

$$0 = \min_{p \in P_{XY}} I_p(\boldsymbol{x}; \boldsymbol{y}) \leq I_{max} \leq \max_{p \in P_{XY}} I_p(\boldsymbol{x}; \boldsymbol{y})$$

there exists a joint distribution $p^*(\boldsymbol{x}, \boldsymbol{y}) \in P_{XY}$ such that

$$I_{p^*}(\boldsymbol{x}; \boldsymbol{y}) = I_{max}$$

As models p_θ and q_ϕ have enough capacity under the constraint in Eq. (5.6), there exist θ^*

and ϕ^* such that $\forall x \in \Sigma_x, \forall y \in \Sigma_y$

$$p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x}) = \frac{p^*(\boldsymbol{x},\boldsymbol{y})}{q(\boldsymbol{x})}$$
$$q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y}) = \frac{p^*(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{y})}$$

thus

$$p_{\theta^*}(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x})q(\boldsymbol{x}) = \sum_{\boldsymbol{x}} p^*(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y})$$
$$q_{\phi^*}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y})p(\boldsymbol{y}) = \sum_{\boldsymbol{y}} p^*(\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{x})$$

and thus

$$\begin{aligned}
I_{p_{\theta^*}} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\mathbb{E}_{\boldsymbol{y}\sim p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x})}\left[\log\frac{p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x})}{p_{\theta^*}(\boldsymbol{y})}\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p^*(\boldsymbol{x},\boldsymbol{y})}\left[\log\frac{p^*(\boldsymbol{x},\boldsymbol{y})}{q(\boldsymbol{x})p(\boldsymbol{y})}\right] \\
&= I_{p^*}(\boldsymbol{x};\boldsymbol{y}) \\
&= I_{max}
\end{aligned}$$

$$\begin{aligned}
I_{q_{\phi^*}} &= \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{y})}\left[\mathbb{E}_{\boldsymbol{x}\sim q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y})}\left[\log\frac{q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y})}{q_{\phi^*}(\boldsymbol{x})}\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim p^*(\boldsymbol{x},\boldsymbol{y})}\left[\log\frac{p^*(\boldsymbol{x},\boldsymbol{y})}{q(\boldsymbol{x})p(\boldsymbol{y})}\right] \\
&= I_{p^*}(\boldsymbol{x};\boldsymbol{y}) \\
&= I_{max}
\end{aligned}$$

and

$$\frac{q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y})}{q_{\phi^*}(\boldsymbol{x})}p(\boldsymbol{y}) = \frac{p^*(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{y})}\frac{p(\boldsymbol{y})}{q(\boldsymbol{x})} = p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x})$$

$$\frac{p_{\theta^*}(\boldsymbol{y}\,|\,\boldsymbol{x})}{p_{\theta^*}(\boldsymbol{y})}q(\boldsymbol{x}) = \frac{p^*(\boldsymbol{x},\boldsymbol{y})}{q(\boldsymbol{x})}\frac{q(\boldsymbol{x})}{p(\boldsymbol{y})} = q_{\phi^*}(\boldsymbol{x}\,|\,\boldsymbol{y})$$

concluding the proof. □

Finally, we connect the global optimum of the dual reconstruction objective to that of the marginal likelihood objective by first proving Lemma 1 and then Theorem 1.

Lemma 1. Let $p(x)$ and $p'(x)$ be two discrete probability functions over random variable $x \in \Sigma_x$, and $q(y)$ and $q'(y)$ be two discrete probability functions over random variable $y \in \Sigma_y$. If $\forall x \in \Sigma_x, \forall y \in \Sigma_y$, $p'(x)q'(y) = p(x)q(y)$, then $\forall x \in \Sigma_x, \forall y \in \Sigma_y$, $p'(x) = p(x)$ and $q'(y) = q(y)$.

Proof. Let $x_0 \in \Sigma_x$ such that $p(x_0) \neq 0$ and $p'(x_0) \neq 0$, and $y_0 \in \Sigma_y$ such that $q(y_0) \neq 0$ and $q'(y_0) \neq 0$.

Since

$$p'(x_0)q'(y_0) = p(x_0)q(y_0) \tag{5.8}$$

and for any $x \in \Sigma_x$

$$p'(x)q'(y_0) = p(x)q(y_0) \tag{5.9}$$

we have

$$\frac{p'(x)}{p'(x_0)} = \frac{p(x)}{p(x_0)} \tag{5.10}$$

.

Given $\sum_{x\in\Sigma_x} p'(x) = 1$, $\sum_{x\in\Sigma_x} p(x) = 1$, and since

$$\sum_{x\in\Sigma_x} p'(x) = \frac{p'(x_0)}{p(x_0)} \sum_{x\in\Sigma_x} p(x) \tag{5.11}$$

we have $p'(x_0) = p(x_0)$.

Thus for any $x \in \Sigma_x$, $p'(x) = \frac{p'(x_0)}{p(x_0)} p(x) = p(x)$. For any $y \in \Sigma_y$, $q'(y) = \frac{p(x)}{p'(x)} q(y) = q(y)$, concluding the proof. □

Theorem 1. Given prior distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ over $\boldsymbol{x} \in \Sigma_x$ and $\boldsymbol{y} \in \Sigma_y$, if parameterized probability models p_θ and q_ϕ have enough capacity under the constraint that:

$$0 \leq I_{min} \leq I_{p_\theta}, I_{q_\phi} \leq I_{max} \leq \max_{p\in P_{XY}} I_p(\boldsymbol{x};\boldsymbol{y})$$

where I_{min} and I_{max} are pre-defined constant values between zero and the maximum mutual information between $\boldsymbol{x}$ and $\boldsymbol{y}$ given any joint distribution $p(\boldsymbol{x},\boldsymbol{y}) \in P_{XY}$ whose marginals satisfy $\sum_{\boldsymbol{x}} p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{y})$ and $\sum_{\boldsymbol{y}} p(\boldsymbol{x},\boldsymbol{y}) = q(\boldsymbol{x})$. Let θ^*, ϕ^* be the global optimum of the dual reconstruction objective $\max_{\theta,\phi} \mathscr{J}_{dual}(\theta,\phi)$, then $q_{\phi^*}(\boldsymbol{x}) = q(\boldsymbol{x})$, $p_{\theta^*}(\boldsymbol{y}) = p(\boldsymbol{y})$, and $I_{q_{\phi^*}} = I_{p_{\theta^*}} = I_{max}$.

Proof. Suppose models p_θ and q_ϕ have enough capacity under the constraint that:

$$0 \leq I_{min} \leq I_{p_\theta}, I_{q_\phi} \leq I_{max} \leq \max_{p\in P_{XY}} I_p(\boldsymbol{x};\boldsymbol{y})$$

then based on Proposition 1, $\mathscr{J}_{dual}(\theta,\phi) \leq 2I_{max} - \mathbb{H}[q(\boldsymbol{x})] - \mathbb{H}[p(\boldsymbol{y})]$, and the upper bound is achieved iff the optimal criteria Eq. (5.5) hold. And based on Proposition 2, there exists a

solution p_{θ^*} an q_{ϕ^*} for the criteria Eq. (5.5). Thus

$$\max_{\theta,\phi} \mathscr{J}_{dual}(\theta,\phi) = 2I_{max} - \mathbb{H}\left[q(\boldsymbol{x})\right] - \mathbb{H}\left[p(\boldsymbol{y})\right]$$

Based on the first equation in Eq. (5.5), we have:

$$I_{q_{\phi^*}} = I_{p_{\theta^*}} = I_{max} \tag{5.12}$$

And multiply the last two equations, we have:

$$p(\boldsymbol{y})q(\boldsymbol{x}) = q_{\phi^*}(\boldsymbol{x})p_{\theta^*}(\boldsymbol{y}) \tag{5.13}$$

Given Lemma 1, we have $q_{\phi^*}(\boldsymbol{x}) = q(\boldsymbol{x})$ and $p_{\theta^*}(\boldsymbol{y}) = p(\boldsymbol{y})$, concluding the proof. □

Thus, while the marginal likelihood objective provides no guarantee for the model's mutual information, the global optimum of dual reconstruction objective guarantees that the mutual information of translation models $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{x}|\boldsymbol{y})$ will be maximized to I_{max}.

5.2.4 Practical Approximations

Despite its desirable optimum, the dual reconstruction objective cannot be directly optimized since decoding is not differentiable. We compare how it is approximated by IBT vs. Dual Learning.

Gradient Approximation To estimate the dual reconstruction objective, one could use sampling or beam search from the model distribution. However, since neither approach is differentiable,

the gradients $\nabla_\theta \mathscr{J}_2$ and $\nabla_\phi \mathscr{J}_1$ cannot be computed directly. IBT blocks the gradients $\nabla_\theta \mathscr{J}_2$ and $\nabla_\phi \mathscr{J}_1$ assuming that they are negligible, while Dual Learning approximates them by policy gradient (Williams, 1992), which can lead to slow and unstable training (Henderson et al., 2018; Wu et al., 2018). Proposition 1 shows that the objective is maximized when the mutual information is maximized to I_{max}. Thus, maximizing the mutual information by other means can help side-step this issue. For example, combining the supervised and unsupervised training objectives (Cotterell and Kreutzer, 2018; Sennrich et al., 2016b) to train models jointly on the parallel and monolingual data can help. For unsupervised MT, the denoising auto-encoding objective introduced in Lample et al. (2018) can be viewed as a way to maximize the mutual information.

LM Loss Dual Learning combines the dual reconstruction objective with an LM loss to encourage the generated translations to be close to the target language domain. Theorem 1 suggests that the LM loss is redundant: optimizing the dual reconstruction objective implicitly pushes the output distributions of the source-to-target and target-to-source models toward the target and source language distributions respectively, which has the same effect intended by the LM loss.

Optimization Strategy While Dual Learning uses batch-level updates, where back-translations are generated on-the-fly and the translation models p_θ and q_ϕ are updated alternately in data batches, IBT adopts different strategies based on the data settings. Batch-level IBT is used in unsupervised MT to quickly boost the model performance from a cold start (Artetxe et al., 2018; Lample et al., 2018), while epoch-level IBT is used in semi-supervised MT, where a fixed

Task	Lang.	Parallel Data		Mono. Data		Validation	Test
high-resource	de-en	News	4.5M	News	5.0M	newstest15	newstest16-18
low-resource	tr-en	News	0.2M	News	0.8M	newstest16	newstest17-18
cross-domain	de-en	News	4.5M	TED	0.5M	iwslt-test14	iwslt-test15-17

Table 5.1: The empirical comparison spans three data conditions (and both translation directions). We report provenance and the number of sentences in parallel and monolingual training data, as well as validation and test sets for each setting. Monolingual data are randomly selected from *"News Crawl: articles from 2015"* for German↔English and *"News Crawl: articles from 2017"* for Turkish↔English, and TED talks data for TED.

model p_θ is used to back-translate the entire monolingual corpus to train q_ϕ until convergence and vice-versa for p_θ (Zhang et al., 2018).

Summary This theoretical analysis suggests that the dual reconstruction objective is a good alternative to the intractable marginal likelihood objective, and that IBT approximates it more closely than the more complex Dual Learning objective. However, we do not know whether the Dual Reconstruction optimum is reached in practice. We therefore conduct an extensive empirical study to determine whether the differences in approximations made by IBT and Dual Learning matter.

5.3 Empirical Study

We evaluate on six translation tasks including German↔English (de-en),[3] Turkish↔English (tr-en) from WMT18 (Bojar et al., 2018),[4] and a cross-domain task which tests de↔en models trained on WMT data on the TED test sets from IWSLT17 (Cettolo et al., 2017).[5]

For preprocessing, we normalize punctuations and apply tokenization, true-casing, and joint source-target Byte Pair Encoding (Sennrich et al., 2016c) with 32,000 operations. We set

[3]We exclude Rapid and ParaCrawl corpora as they are noisy and thus require data filtering (Morishita et al., 2018).

[4]http://www.statmt.org/wmt18/translation-task.html

[5]https://wit3.fbk.eu/mt.php?release=2017-01-ted-test

the maximum sentence length to 50.

5.3.1 Model and Training Configuration

We adopt the base Transformer model (Vaswani et al., 2017) with $d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $n_{\text{heads}} = 8$, $n_{\text{layers}} = 6$, and $p_{\text{drop}} = 0.1$. We tie the source and target embeddings with the output layer weights (Nguyen and Chiang, 2018; Press and Wolf, 2017). We pre-train models with the supervised objective until convergence, and fine-tune on the mixed parallel and monolingual data as in prior work (Cotterell and Kreutzer, 2018; Sennrich et al., 2016b). We use the Adam optimizer (Kingma and Ba, 2015) with a batch size of 32 sentences and checkpoint the model every 2500 updates. Training hyperparameters and stopping criteria are constant across all comparable experimental conditions. Initial learning rates for pre-training and fine-tuning are respectively set to 10^{-4} and 2×10^{-5}. We decay the learning rate by 30% and reload the best model after 3 checkpoints without improvement. We apply early stopping after repeating this process for 5 times. We adopt the same learning rate decay and stopping criteria during fine-tuning. For batch-level IBT and Dual Learning, we check whether both models improve validation perplexity. For epoch-level IBT, we run for 3 iterations. The LMs in Dual Learning are RNNs (Mikolov et al., 2010a) with 512 hidden units, embeddings of size 512, and dropout of 0.2 to hidden states. We tie the input embeddings with the output layer weights. We clip the gradients at a threshold of 5. We train them similarly to NMT models, except for setting the batch size to 64 sentences and the initial learning rate to 0.001. We decay the learning rate by 50% and reload the best model after 5 checkpoints without validation perplexity improvement and apply early stopping after repeating the process for 5 times. We report the validation perplexity of the NMT

and LM models in Table 5.2, and the model sizes in Table 5.3. All experiments are performed on a single NVIDIA GeForce GTX 1080 Ti GPU.

At decoding time, we use beam search with a beam size of 5.

Task	NMT.xx-en	NMT.en-xx	LM.xx	LM.en
low-resource	22.12	30.92	121.13	78.96
high-resource	6.72	6.25	78.32	74.00
cross-domain	8.17	7.35	102.43	92.70

Table 5.2: Validation perplexity of the NMT and LM models. We denote English as *en* and the other language as *xx*.

Task	NMT.xx-en	NMT.en-xx	LM.xx	LM.en
low-resource	92811565	92811565	27311123	19247448
high-resource	98346302	98346302	32165523	24095698
cross-domain	98346302	98346302	24806023	18768773

Table 5.3: Number of model parameters. We denote English as *en* and the other language as *xx*.

5.3.2 Baselines and Evaluation

Our experiments are based on strong supervised baselines.[6] We compare semi-supervised models that are fine-tuned with Back-Translation, epoch-level and batch-level IBT, and Dual Learning with varying interpolation weights $\alpha_{LM} = \{0, 0.1, 0.5\}$ for the LM loss.[7] Following He et al. (2016), we use beam search with a beam size of 2 for inference in Dual Learning and IBT.

We evaluate translation quality using sacreBLEU[8] and total training time in hours. We also show learning curves for the approximated dual reconstruction loss (negative of the dual reconstruction objective in Eq. (5.2), averaged over the training batches from both directions).

[6] de-en: 2–4 BLEU higher than the baseline of Morishita et al. (2018); tr-en: on par or higher than the baseline of García-Martínez et al. (2017).

[7] By contrast, prior work only reports results for $\alpha_{LM} = 0.005$ (He et al., 2016). Our preliminary result show that $\alpha_{LM} = 0.005$ obtains similar results to $\alpha_{LM} = 0$.

[8] Version: BLEU +case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.2.11

Low-Resource	α_{LM}	hours	tr-en BLEU			en-tr BLEU		
			2017	2018	Avg	2017	2018	Avg
baseline	–	8.0	15.14	15.95	15.55	11.17	10.18	10.68
epoch-level IBT-1	–	86.1	16.36	16.44	16.40	15.08	12.98	**14.03**
epoch-level IBT-2	–	162.2	19.12	19.63	**19.38**	14.94	12.53	13.74
epoch-level IBT-3	–	237.5	18.76	19.01	18.89	15.04	12.93	13.99
batch-level IBT	–	160.6	17.18	18.08	17.63	13.90	11.84	12.87
Dual Learning	0.0	313.2	17.07	18.00	17.54	14.17	11.91	13.04
Dual Learning	0.1	257.8	17.09	17.62	17.36	13.88	11.49	12.69
Dual Learning	0.5	421.2	17.33	18.36	17.85	14.54	12.30	13.42

High-Resource	α_{LM}	hours	de-en BLEU				en-de BLEU			
			2016	2017	2018	Avg	2016	2017	2018	Avg
baseline	–	26.7	31.95	27.74	34.59	31.43	29.18	23.46	34.53	29.06
epoch-level IBT-1	–	439.0	32.59	28.46	35.22	32.09	30.13	23.87	35.35	29.78
epoch-level IBT-2	–	850.9	33.64	29.13	36.37	**33.05**	29.99	24.42	35.60	**30.00**
epoch-level IBT-3	–	1261.6	33.43	29.07	36.17	32.89	29.93	24.24	35.46	29.88
batch-level IBT	–	94.0	32.95	28.65	35.24	32.28	29.70	23.78	34.89	29.46
Dual Learning	0.0	128.2	32.79	28.47	35.10	32.12	29.37	23.50	34.67	29.18
Dual Learning	0.1	93.3	32.63	28.47	34.88	31.99	29.38	23.79	34.71	29.29
Dual Learning	0.5	152.1	32.89	28.69	35.32	32.30	29.58	23.65	34.88	29.37

Cross-Domain	α_{LM}	hours	de-en BLEU				en-de BLEU			
			2015	2016	2017	Avg	2015	2016	2017	Avg
baseline	–	26.2	27.11	27.37	23.65	26.04	26.35	23.10	21.69	23.71
epoch-level IBT-1	–	71.1	28.88	28.73	25.37	27.66	26.69	24.02	22.59	24.43
epoch-level IBT-2	–	115.0	28.70	28.72	25.37	27.60	27.57	24.50	22.78	**24.95**
epoch-level IBT-3	–	159.8	29.13	29.00	25.33	**27.82**	27.31	24.37	22.92	24.87
batch-level IBT	–	45.0	28.03	27.78	24.53	26.78	26.84	23.64	22.35	24.28
Dual Learning	0.0	65.8	28.04	27.73	24.36	26.71	26.70	23.85	22.21	24.25
Dual Learning	0.1	59.3	27.77	27.84	24.51	26.71	26.99	23.86	22.59	24.48
Dual Learning	0.5	92.7	27.84	28.00	24.18	26.67	27.23	24.08	22.72	24.68

Table 5.4: BLEU scores and total training time (*hours*) on the low-resource, high-resource, and cross-domain tasks. *epoch-level* IBT-1, IBT-2, and IBR-3 denotes models fine-tuned with IBT for 1–3 iterations, and α_{LM} denotes the weight for the LM loss. We boldface the highest average scores. Overall, epoch-level IBT outperforms all other methods at the cost of much longer training time.

5.3.3 Findings

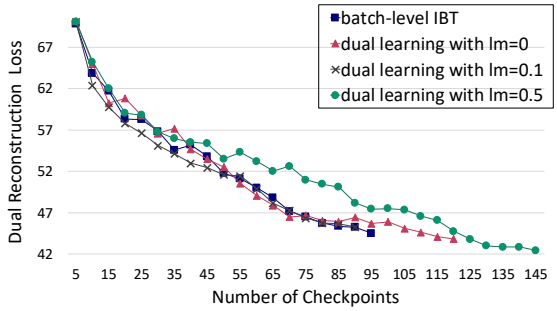
Overview All semi-supervised training techniques improve translation quality over the supervised-only baseline (Table 5.4). The first iteration of IBT (i.e. Back-Translation) on monolingual data improves over the baseline by 0.7–3.4 BLEU. IBT is more effective in the direction where the model in the opposite direction is most improved by Back-Translation For example, in the high and low resource tasks where Back-Translation improves over the baseline more when translating out of English, the best performing IBT model improves BLEU greatly over Back-Translation when translating into English (+1.0–3.0 BLEU), but not in the other direction. In the cross-domain scenario where Back-Translation improves more on de-en, IBT outperforms Back-Translation more on en-de than the other direction.

Impact of Policy Gradient Updating the inference model via policy gradient fails to lower the dual reconstruction loss and has little impact on BLEU. We compare Dual Learning (with $\alpha_{LM} = 0$) to batch-level IBT, so that the only difference between the two approaches is whether the inference model is updated. Batch-level IBT achieves similar or higher BLEU than Dual Learning for all tasks, except for the low-resource en-tr task where the BLEU difference is small (< 0.2). In addition, batch-level IBT trains 30–50% faster than Dual Learning. Figure 5.2 shows that the policy gradient update has little impact on the dual reconstruction loss on all tasks.

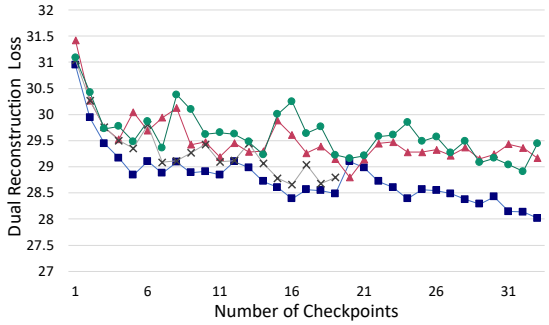
Impact of LM The best Dual Learning BLEU is obtained with $\alpha_{LM} = 0.5$ on all tasks except for de-en in the cross-domain setting (Table 5.4). However, it brings only small BLEU improvements (0.2–0.4) over Dual Learning without LM loss ($\alpha_{LM} > 0$), but causes the dual reconstruction loss to decrease slower (Figure 5.2), and slows down training by 20–40%. In all

cases, IBT outperforms Dual Learning.

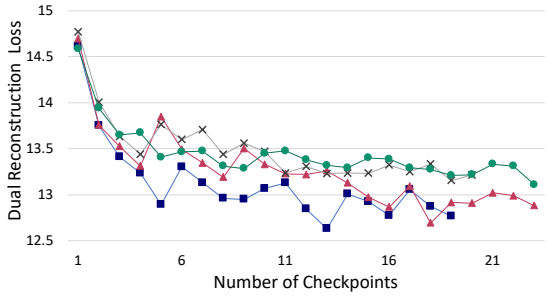
Epoch vs. Batch IBT The best epoch-level IBT model outperforms batch-level IBT by 0.5–1.8 BLEU overall, at the cost of much slower training: 13 times longer in the high-resource setting, 1.5 times longer in the low-resource setting, and 3.5 times longer in the cross-domain setting. Running IBT for two iterations is a good choice to balance training efficiency and translation quality, as the third iteration does not help BLEU.



(a) Low-resource task



(b) High-resource task



(c) Cross-domain task

Figure 5.2: Learning curves for the approximated dual reconstruction loss averaged over the training batches from both directions on the low-resource, high-resource, and cross-domain tasks.

5.3.4 Mutual Information Analysis

We test the hypothesis that the mutual information constraint is met when training models on the combined supervised and unsupervised objectives in the low-resource setting (the most adversarial condition with the fewest supervised training samples).

The mutual information I_{p_θ} from Definition 5.2 can be computed by Hoffman and Johnson (2016):

$$\begin{aligned} I_{p_\theta} =& \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left[\, p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x}) || \, p(\boldsymbol{y})\right]\right] \\ & - D_{\mathrm{KL}}\left[\, p_\theta(\boldsymbol{y}) || \, p(\boldsymbol{y})\right] \end{aligned} \tag{5.14}$$

where prior distributions $q(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are estimated by the empirical data distribution given the monolingual corpora $\mathscr{M}_X$ and $\mathscr{M}_Y$. Although computing I_{p_θ} directly is intractable, it can be approximated with a Monte Carlo estimate. Following Dieng et al. (2019), we approximate the two KL terms by Monte Carlo, where samples from $p_\theta(\boldsymbol{y})$ can be obtained by ancestral sampling (we use beam search with beam size of five to sample from $p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x})$). The marginal probability $p_\theta(\boldsymbol{y}) = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x})\right]$ can also be estimated by Monte Carlo. Due to data sparsity, the conditional likelihood $p_\theta(\boldsymbol{y}\,|\,\boldsymbol{x})$ will be near zero for most source sentences randomly sampled from $q(\boldsymbol{x})$. To better estimate it, we smooth the data distribution of the original dataset $\mathscr{D}$ by generating a randomly perturbed dataset $\tilde{\mathscr{D}}$.[9]

Table 5.5 shows the normalized mutual information $\tilde{I} - \log|\mathscr{D}|$ where $\tilde{I}$ denotes the estimated mutual information. It shows that, when training with the combination of supervised

[9]We generate 20 perturbed sentences per source via random word dropping with probability of 0.1 and permutation with maximum distance of 3.

	tr-en	en-tr
baseline	-2.47	-2.28
epoch-level IBT-1	-2.57	-2.23
epoch-level IBT-2	-2.18	-2.30
epoch-level IBT-3	-2.32	-2.42
batch-level IBT	-1.51	-1.82
dual learning w/ $\alpha_{LM} = 0$	-1.50	-1.80
dual learning w/ $\alpha_{LM} = 0.5$	-1.44	-1.77

Table 5.5: Results on estimated mutual information $\tilde{I}$ in the low-resource setting. We report the normalized scores $\tilde{I} - \log|\mathscr{D}|$ (on the scale of 10^{-4}) averaged over the two test sets. The range of normalized scores should be $[-\log|\mathscr{D}|, \log\frac{|\tilde{\mathscr{D}}|}{|\mathscr{D}|}] = [-8.0, 3.0]$.

and unsupervised objectives, the normalized mutual information is within a small range between $(-2.6 \times 10^{-4}, -1.4 \times 10^{-4})$ and is lower than the maximum normalized mutual information $\log|\tilde{\mathscr{D}}| - \log|\mathscr{D}| \approx 3.0$ by a large margin. Thus, the mutual information can be bounded by appropriate values of I_{min} and I_{max} to satisfy the constraint. In addition, these results confirm that updating the inference model using policy gradient in Dual Learning does not effectively increase model's mutual information.

5.3.5 Limitations

The findings of this work should be interpreted with several limitations in mind. First, our empirical study is limited to two language pairs in three data settings. Further experiments are needed to examine how the findings generalize to other language pairs (especially the extremely low-resource languages) and data settings. In addition, we use BLEU as the quality metric for the empirical study, while it remains unexplored how the models compare based on human judgements.

5.4 Summary

We contributed a theoretical framework, namely Dual Reconstruction Objective, that unifies and theoretically grounds the comparison of existing semi-supervised learning approaches for NMT including Back-Translation, Iterative Back-Translation (IBT) and Dual Learning.

On the theory side, we defined a dual reconstruction objective which unifies semi-supervised NMT techniques that exploit source and target monolingual text. We proved that optimizing this objective leads to the same global optimum as the intractable marginal likelihood objective, where the model's marginal distribution coincides with the prior language distribution while also maximizing the model's mutual information between source and target. IBT approximates this objective more closely than Dual Learning, despite the more complex objective and update strategies used in the latter.

We presented a systematic empirical comparison of Back-Translation, IBT, and Dual Learning on six tasks spanning high-resource, low-resource, and cross-domain settings. Results support the theory that the LM loss and policy gradient estimation are unnecessary in Dual Learning, and show that IBT achieves better translation quality than Dual Learning. Analysis confirms that in practice, the mutual information constraint required to reach an interesting dual reconstruction optimum is satisfied through the joint optimization of the supervised and unsupervised objectives.

This work sheds new lights on the inductive biases in semi-supervised learning algorithms for effective integration of prior language distributions in NMT. In the next chapter, we change our focus to another form of inductive bias, i.e. more controllable model architectures that allow users to inject human knowledge into NMT models more easily.

Chapter 6: Stronger Inductive Biases for Controllable Machine Translation

6.1 Introduction

Having demonstrated the effectiveness of integrating stronger inductive biases through training algorithms, we then ask if we can strengthen the inductive biases in NMT through more controllable model architectures and modular frameworks to allow users to inject knowledge and communicate their intention to the model. Neural machine translation (MT) architectures (Bahdanau et al., 2015; Vaswani et al., 2017) typically share the same inductive bias that a translation sequence should be predicted token by token from left to right, which makes it difficult for users to specify preferences that could be incorporated more easily in statistical MT models (Koehn et al., 2007) and have been shown to be useful for interactive machine translation (Barrachina et al., 2009; Foster et al., 2002) and domain adaptation (Hokamp and Liu, 2017). Lexical constraints or preferences have previously been incorporated by re-training NMT models with constraints as inputs (Dinu et al., 2019; Song et al., 2019) or with constrained beam search that drastically slows down decoding (Hokamp and Liu, 2017; Post and Vilar, 2018).

In this work, we introduce a translation model that can seamlessly incorporate users' lexical choice preferences without increasing the time and computational cost at decoding time, while being trained on regular MT samples. We apply this model to MT tasks with soft lexical constraints. As illustrated in Figure 6.1, when decoding with soft lexical constraints,

source Jucătorul de 29 de ani sa luptat doi ani cu problemele la gleznă.
reference The 29-year-old has been plagued with a troublesome ankle for two years.
constraints: plague ankle
unconstrained MT output The 29-year-old has struggled for two years with problems in the bullying.
hard-constrained MT output The 29-year-old has been plague for two years with problems in the ankle.
soft-constrained MT output The 29-year-old has struggled for two years with problems in the ankle.

Figure 6.1: Romanian to English MT example. Unconstrained MT incorrectly translates "gleznă" to "bullying". Given constraint words "plague" and "ankle", soft-constrained MT correctly uses "ankle" and avoids disfluencies introduced by using "plague" as a hard constraint in its exact form.

user preferences for lexical choice in the output language are provided as an additional input sequence of target words in any order. The goal is to let users encode terminology, domain or stylistic preferences in target word usage, without strictly enforcing hard constraints that might hamper NMT's ability to generate fluent outputs.

Our model is an **Edi**t-Based **T**ransf**O**rmer with **R**epositioning (**EDITOR**), which builds on recent progress on non-autoregressive sequence generation (Ghazvininejad et al., 2019; Lee et al., 2018a).[1] Specifically, the Levenshtein Transformer (Gu et al., 2019) showed that iteratively refining output sequences via insertions and deletions yields a fast and flexible generation process for MT and automatic post-editing tasks. EDITOR replaces the deletion operation with a novel reposition operation to disentangle lexical choice from reordering decisions. As a result, EDITOR exploits lexical constraints more effectively and efficiently than the Levenshtein Transformer, as a single reposition operation can subsume a sequence of deletions and insertions. To train EDITOR via imitation learning, the reposition operation is defined to preserve the ability to use

[1] https://github.com/Izecson/fairseq-editor

the Levenshtein edit distance (Levenshtein, 1966) as an efficient oracle. We also introduce a dual-path roll-in policy which lets the reposition and deletion models learn to refine their respective outputs more effectively.

Experiments on Romanian-English, English-German, and English-Japanese MT show that EDITOR achieves comparable or better translation quality with faster decoding speed than the Levenshtein Transformer (Gu et al., 2019) on the standard MT tasks and exploit soft lexical constraints better: it achieves significantly better translation quality and matches more constraints with faster decoding speed than the Levenshtein Transformer. It also drastically speeds up decoding compared to lexically constrained decoding algorithms (Post and Vilar, 2018). Furthermore, results highlight the benefits of soft constraints over hard ones – EDITOR with soft constraints achieves translation quality on par or better than both EDITOR and Levenshtein Transformer with hard constraints (Susanto et al., 2020).

6.2 EDITOR

6.2.1 Model

We cast both constrained and unconstrained language generation as an iterative sequence refinement problem modeled by a Markov Decision Process $(\mathscr{Y}, \mathscr{A}, \mathscr{E}, \mathscr{R}, \mathbf{y}^0)$, where a state $\mathbf{y}$ in the state space $\mathscr{Y}$ corresponds to a sequence of tokens $\mathbf{y} = (y_1, y_2, ..., y_L)$ from the vocabulary $\mathscr{V}$ up to length L, and $\mathbf{y}^0 \in \mathscr{Y}$ is the initial sequence For standard sequence generation tasks, $\mathbf{y}^0$ is the empty sequence $(\langle s \rangle, \langle /s \rangle)$. For lexically constrained generation tasks, $\mathbf{y}^0$ consists of the words to be used as constraints $(\langle s \rangle, c_1, ..., c_m, \langle /s \rangle)$.

At the k-th decoding iteration, the model takes as input $\mathbf{y}^{k-1}$, the output from the previous

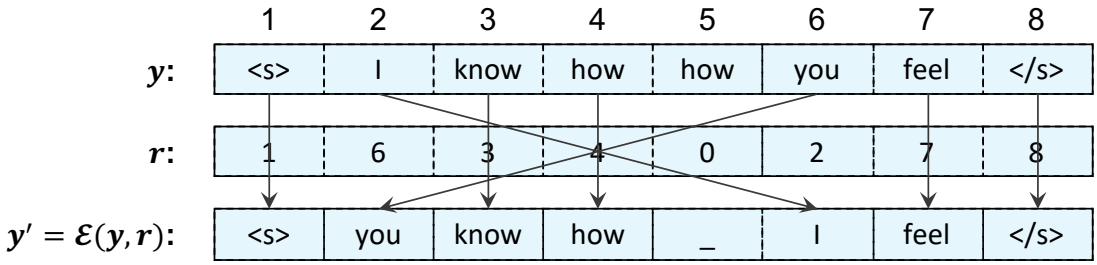


Figure 6.2: Applying the reposition operation $\boldsymbol{r}$ to input $\boldsymbol{y}$: $r_i > 0$ is the 1-based index of token y'_i in the input sequence; y_i is deleted if $r_i = 0$.

iteration, chooses an action $\boldsymbol{a}^k \in \mathscr{A}$ to refine the sequence into $\boldsymbol{y}^k = \mathscr{E}(\boldsymbol{y}^{k-1}, \boldsymbol{a}^k)$, and receives a reward $r^k = \mathscr{R}(\boldsymbol{y}^k)$. The policy π maps the input sequence $\boldsymbol{y}^{k-1}$ to a probability distribution $P(\mathscr{A})$ over the action space $\mathscr{A}$. Our model is based on the Transformer encoder-decoder (Vaswani et al., 2017) and we extract the decoder representations $(\boldsymbol{h}_1, ..., \boldsymbol{h}_n)$ to make the policy predictions. Each refinement action is based on two basic operations: reposition and insertion.

Reposition For each position i in the input sequence $\boldsymbol{y}_{1...n}$, the reposition policy $\pi_{rps}(r \,|\, i, \boldsymbol{y})$ predicts an index $r \in [0, n]$: if $r > 0$, we place the r-th input token y_r at the i-th output position, otherwise we delete the token at that position (Figure 6.2). We constrain $\pi_{rps}(1 \,|\, 1, \boldsymbol{y}) = \pi_{rps}(n \,|\, n, \boldsymbol{y}) = 1$ to maintain sequence boundaries. Note that reposition differs from typical reordering since 1) it makes it possible to delete tokens, and 2) it places tokens at each position independently, which enables parallelization at decoding time. In principle, the same input token can thus be placed at multiple output positions. However, this happens rarely in practice as the policy predictor is trained to follow oracle demonstrations which cannot contain such repetitions by design.[2]

The reposition classifier gives a categorical distribution over the index of the input token to

[2]Empirically, fewer than 1% of tokens are repositioned to more than one output position.

be placed at each output position:

$$\pi_{rps}(r \mid i, \boldsymbol{y}) = \text{softmax}(\boldsymbol{h}_i \cdot [\boldsymbol{b}, \boldsymbol{e}_1, ..., \boldsymbol{e}_n]) \tag{6.1}$$

where $\boldsymbol{e}_j$ is the embedding of the j-th token in the input sequence, and $\boldsymbol{b} \in \mathbb{R}^{d_{model}}$ is used to predict whether to delete the token. The dot product in the softmax function captures the similarity between the hidden state $\boldsymbol{h}_i$ and each input embedding $\boldsymbol{e}_j$ or the deletion vector $\boldsymbol{b}$.

Insertion Following Gu et al. (2019), the insertion operation consists of two phases: (1) *placeholder insertion:* given an input sequence $\boldsymbol{y}_{1...n}$, the placeholder predictor $\pi_{plh}(p \mid i, \boldsymbol{y})$ predicts the number of placeholders $p \in [0, K_{max}]$ to be inserted between two neighboring tokens (y_i, y_{i+1});[3] (2) *token prediction:* given the output of the placeholder predictor, the token predictor $\pi_{tok}(t \mid i, \boldsymbol{y})$ replaces each placeholder with an actual token.

The Placeholder Insertion Classifier gives a categorical distribution over the number of placeholders to be inserted between every two consecutive positions:

$$\pi_{plh}(p \mid i, \boldsymbol{y}) = \text{softmax}([\boldsymbol{h}_i ; \boldsymbol{h}_{i+1}] \cdot \boldsymbol{W}^{plh}) \tag{6.2}$$

where $\boldsymbol{W}^{plh} \in \mathbb{R}^{(2d_{model}) \times (K_{max}+1)}$.

The Token Prediction Classifier predicts the identity of each token to fill in each placeholder:

$$\pi_{tok}(t \mid i, \boldsymbol{y}) = \text{softmax}(\boldsymbol{h}_i \cdot \boldsymbol{W}^{tok}) \tag{6.3}$$

[3] In our implementation, we set $K_{max} = 255$.

where $\boldsymbol{W}^{tok} \in \mathbb{R}^{d_{model} \times |\mathcal{V}|}$.

Action Given an input sequence $\boldsymbol{y}_{1...n}$, an action consists of repositioning tokens, inserting and replacing placeholders. Formally, we define an action as *a sequence* of reposition ($\boldsymbol{r}$), placeholder insertion ($\boldsymbol{p}$), and token prediction ($\boldsymbol{t}$) operations: $\boldsymbol{a} = (\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{t})$. $\boldsymbol{r}$, $\boldsymbol{p}$, and $\boldsymbol{t}$ are applied in this order to adjust non-empty initial sequences via reposition before inserting new tokens. Each of $\boldsymbol{r}$, $\boldsymbol{p}$, and $\boldsymbol{t}$ consists of a set of basic operations that can be applied *in parallel*:

$$\boldsymbol{r} = \{r_1, ..., r_n\}$$

$$\boldsymbol{p} = \{p_1, ..., p_{m-1}\}$$

$$\boldsymbol{t} = \{t_1, ..., t_l\}$$

where $m = \sum_i^n \mathbb{I}(r_i > 0)$ and $l = \sum_i^{m-1} p_i$. We define the policy as

$$\pi(\boldsymbol{a}|\boldsymbol{y}) = \prod_{r_i \in \boldsymbol{r}} \pi_{rps}(r_i \,|\, i, \boldsymbol{y}) \cdot \prod_{p_i \in \boldsymbol{p}} \pi_{plh}(p_i \,|\, i, \boldsymbol{y}') \cdot \prod_{t_i \in \boldsymbol{t}} \pi_{tok}(t_i \,|\, i, \boldsymbol{y}'')$$

with intermediate outputs $\boldsymbol{y}' = \mathscr{E}(\boldsymbol{y}, \boldsymbol{r})$ and $\boldsymbol{y}'' = \mathscr{E}(\boldsymbol{y}', \boldsymbol{p})$.

6.2.2 Dual-Path Imitation Learning

We train EDITOR using imitation learning (Daumé III et al., 2009; Ross and Bagnell, 2014; Ross et al., 2011) to efficiently explore the space of valid action sequences that can reach a reference translation. The key idea is to construct a *roll-in* policy π^{in} to generate sequences to be refined and a *roll-out* policy π^{out} to estimate cost-to-go for all possible actions given each input

sequence. The model is trained to choose actions that minimizes the cost-to-go estimates. We use a search-based oracle policy π^* as the roll-out policy and train the model to imitate the optimal actions chosen by the oracle.

Formally, $\boldsymbol{d}_{\pi^{in}_{rps}}$ and $\boldsymbol{d}_{\pi^{in}_{ins}}$ denote the distributions of sequences induced by running the roll-in policies π^{in}_{rps} and π^{in}_{ins} respectively. We update the model policy $\pi = \pi_{rps} \cdot \pi_{plh} \cdot \pi_{tok}$ to minimize the expected cost $\mathscr{C}(\pi; \boldsymbol{y}, \pi^*)$ by comparing the model policy against the cost-to-go estimates under the oracle policy π^* given input sequences $\boldsymbol{y}$:

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{y}_{rps} \sim \boldsymbol{d}_{\pi^{in}_{rps}}} \left[\mathscr{C}(\pi_{rps}; \boldsymbol{y}_{rps}, \pi^*) \right] + \\ &\mathbb{E}_{\boldsymbol{y}_{ins} \sim \boldsymbol{d}_{\pi^{in}_{ins}}} \left[\mathscr{C}(\pi_{plh}, \pi_{tok}; \boldsymbol{y}_{ins}, \pi^*) \right] \end{aligned} \tag{6.4}$$

The cost function compares the model vs. oracle actions. As prior work suggests that cost functions close to the cross-entropy loss are better suited to deep neural models than the squared error (Cheng et al., 2018; Leblond et al., 2018), we define the cost function as the KL divergence between the action distributions given by the model policy and by the oracle (Welleck et al., 2019):

$$\begin{aligned} &\mathscr{C}(\pi; \boldsymbol{y}, \pi^*) \\ =&D_{\mathrm{KL}} \left[\pi^*(\boldsymbol{a} \,|\, \boldsymbol{y}, \boldsymbol{y}^*) || \, \pi(\boldsymbol{a} \,|\, \boldsymbol{y}) \right] \\ =&\mathbb{E}_{\boldsymbol{a} \sim \pi^*(\boldsymbol{a} \,|\, \boldsymbol{y}, \boldsymbol{y}^*)} \left[-\log \pi(\boldsymbol{a} \,|\, \boldsymbol{y}) \right] + const. \end{aligned} \tag{6.5}$$

where the oracle has additional access to the reference sequence $\boldsymbol{y}^*$. By minimizing the cost function, the model learns to imitate the oracle policy without access to the reference sequence.

Next, we describe how the reposition operation is incorporated in the roll-in

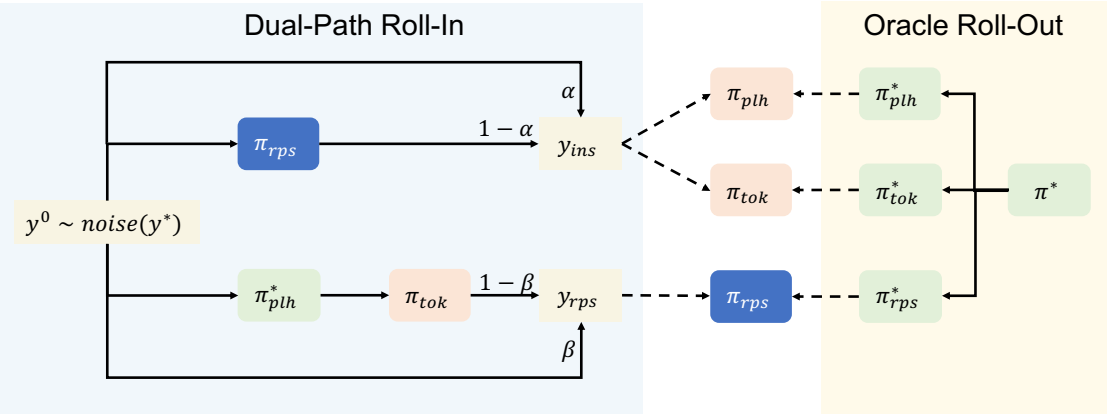


Figure 6.3: Our dual-path imitation learning process uses both the reposition and insertion policies during roll-in so that they can be trained to refine each other's outputs: Given an initial sequence $\boldsymbol{y}^0$, created by noising the reference $\boldsymbol{y}^*$, the roll-in policy stochastically generates intermediate sequences $\boldsymbol{y}_{ins}$ and $\boldsymbol{y}_{rps}$ via reposition and insertion respectively. The policy predictors are trained to minimize the costs of reaching $\boldsymbol{y}^*$ from $\boldsymbol{y}_{ins}$ and $\boldsymbol{y}_{rps}$ estimated by the oracle policy π^*.

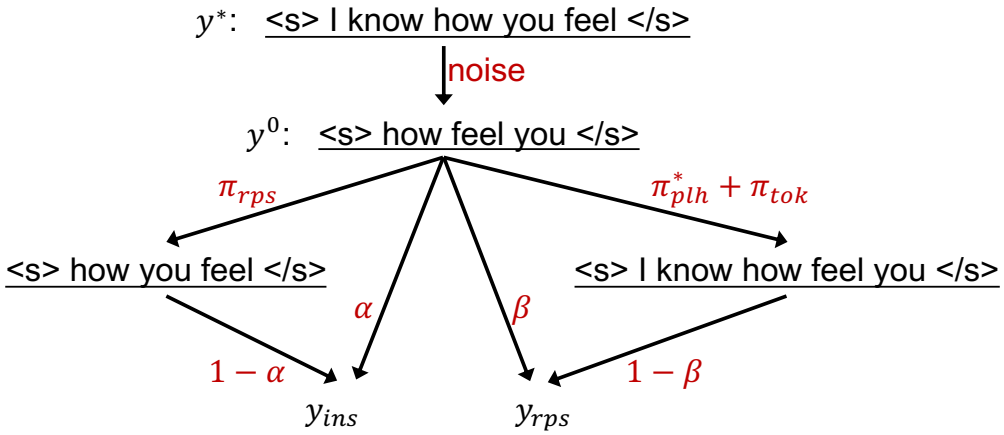


Figure 6.4: The roll-in sequence for the insertion predictor is a stochastic mixture of the noised reference $\boldsymbol{y}^0$ and the output by applying the model's reposition policy π_{rps} to $\boldsymbol{y}^0$. The roll-in sequence for the reposition predictor is a stochastic mixture of the noised reference $\boldsymbol{y}^0$ and the output by applying the oracle placeholder insertion policy π^*_{plh} and the model's token prediction policy π_{tok} to $\boldsymbol{y}^0$.

policy (Section 6.2.2.1) and the oracle roll-out policy (Section 6.2.2.2).

6.2.2.1 Dual-Path Roll-in Policy

As shown in Figure 6.3, the roll-in policies π^{in}_{ins} and π^{in}_{rps} for the reposition and insertion policy predictors are stochastic mixtures of the noised reference sequences and the output sequences sampled from their corresponding dual policy predictors. Figure 6.4 shows an example

for creating the roll-in sequences: we first create the initial sequence $\boldsymbol{y}^0$ by applying random word dropping (Gu et al., 2019) and random word shuffle (Lample et al., 2018) with probability of 0.5 and maximum shuffle distance of 3 to the reference sequence $\boldsymbol{y}^*$, and produce the roll-in sequences for each policy predictor as follows:

1. **Reposition:** the roll-in policy π^{in}_{rps} is a stochastic mixture of the initial sequence $\boldsymbol{y}^0$ and the output sequence by applying one iteration of the oracle placeholder insertion policy $\boldsymbol{p}^* \sim \pi^*$ and the model's token prediction policy $\tilde{\boldsymbol{t}} \sim \pi_{tok}$ to $\boldsymbol{y}^0$:

$$\boldsymbol{d}_{\pi^{in}_{rps}} = \begin{cases} \boldsymbol{y}^0, & \text{if } u < \beta \\ \mathscr{E}(\mathscr{E}(\boldsymbol{y}^0, \boldsymbol{p}^*), \tilde{\boldsymbol{t}}), & \text{otherwise} \end{cases} \tag{6.6}$$

   where the mixture factor $\beta \in [0,1]$ and random variable $u \sim \text{Uniform}(0,1)$.

2. **Insertion:** the roll-in policy π^{in}_{ins} is a stochastic mixture of the initial sequence $\boldsymbol{y}^0$ and the output sequence by applying one iteration of the model's reposition policy $\tilde{\boldsymbol{r}} \sim \pi_{rps}$ to $\boldsymbol{y}^0$:

$$\boldsymbol{d}_{\pi^{in}_{ins}} = \begin{cases} \boldsymbol{y}^0, & \text{if } u < \alpha \\ \mathscr{E}(\boldsymbol{y}^0, \tilde{\boldsymbol{r}}), & \text{otherwise} \end{cases} \tag{6.7}$$

   where the mixture factor $\alpha \in [0,1]$ and random variable $u \sim \text{Uniform}(0,1)$.

While Gu et al. (2019) define roll-in using only the model's insertion policy, we call our approach dual-path because roll-in creates two distinct intermediate sequences using the model's reposition or insertion policy. This makes it possible for the reposition and insertion policy predictors to learn to refine one another's outputs during roll-out, mimicking the iterative

refinement process used at inference time.[4]

6.2.2.2 Oracle Roll-Out Policy

Policy Given an input sequence $\boldsymbol{y}$ and a reference sequence $\boldsymbol{y}^*$, the oracle algorithm finds the optimal action to transform $\boldsymbol{y}$ into $\boldsymbol{y}^*$ with the minimum number of basic edit operations:

$$\text{Oracle}(\boldsymbol{y}, \boldsymbol{y}^*) = \arg\min_{\boldsymbol{a}} \text{NumOps}(\boldsymbol{y}, \boldsymbol{y}^* \,|\, \boldsymbol{a}) \tag{6.8}$$

The associated oracle policy is defined as:

$$\pi^*(\boldsymbol{a} \,|\, \boldsymbol{y}, \boldsymbol{y}^*) = \begin{cases} 1, & \text{if } \boldsymbol{a} = \text{Oracle}(\boldsymbol{y}, \boldsymbol{y}^*) \\ 0, & \text{otherwise} \end{cases} \tag{6.9}$$

Algorithm The reposition and insertion operations used in EDITOR are designed so that the Levenshtein edit distance algorithm (Levenshtein, 1966) can be used as the oracle. The reposition operation (Section 6.2.1) can be split into two distinct types of operations: (1) deletion and (2) replacing a word with any other word appearing in the input sequence, which is a constrained version of the Levenshtein substitution operation. As a result, we can use dynamic programming to find the optimal action sequence in $O(|\boldsymbol{y}||\boldsymbol{y}^*|)$ time. By contrast, the Levenshtein Transformer restricts the oracle and model to insertion and deletion operations only. While in principle substitutions can be performed indirectly by deletion and re-insertion, our results show the benefits of using the reposition variant of the substitution operation.

[4]Different from the inference process, we generate the roll-in sequences by applying the model's reposition or insertion policy for only one iteration.

6.2.3 Inference

During inference, we start from the initial sequence $\boldsymbol{y}^0$. For standard sequence generation tasks, $\boldsymbol{y}^0$ is an empty sequence, whereas for lexically constrained generation $\boldsymbol{y}^0$ is a sequence of lexical constraints. Inference then proceeds in the exact same way for constrained and unconstrained tasks. The initial sequence is refined iteratively by applying a sequence of actions $(\boldsymbol{a}^1, \boldsymbol{a}^2, ...) = (\boldsymbol{r}^1, \boldsymbol{p}^1, \boldsymbol{t}^1; \boldsymbol{r}^2, \boldsymbol{p}^2, \boldsymbol{t}^2; ...)$. We greedily select the best action at each iteration given the model policy in Eqs. (6.1) to (6.3). We stop refining if 1) the output sequences from two consecutive iterations are the same (Gu et al., 2019), or 2) the maximum number of decoding steps is reached (Ghazvininejad et al., 2019; Lee et al., 2018a).[5]

Incorporating Soft Constraints Although EDITOR is trained without lexical constraints, it can be used seamlessly for MT with constraints without any change to the decoding process except using the constraint sequence as the initial sequence.

Incorporating Hard Constraints We adopt the decoding technique introduced by Susanto et al. (2020) to enforce hard constraints at decoding time by prohibiting deletion operations on constraint tokens or insertions within a multi-token constraints.

6.3 Experiments

We evaluate the EDITOR model on standard (Section 6.3.2) and lexically constrained machine translation (Sections 6.3.3–6.3.4).

[5]Following Stern et al. (2019), we also experiment with adding penalty for inserting "empty" placeholders during inference by subtracting a penalty score $\gamma = [0, 3]$ from the logits of zero in Eq. (6.2) to avoid overly short outputs. However, preliminary experiments show that zero penalty score achieves the best performance.

	Train	Valid	Test	Provenance
Ro-En	599k	1911	1999	WMT16
En-De	3,961k	3000	3003	WMT14
En-Ja	2,000k	1790	1812	WAT2017

Table 6.1: MT Tasks: data statistics (# sentence pairs) and provenance per language pair.

6.3.1 Experimental Settings

Dataset Following Gu et al. (2019), we experiment on three language pairs spanning different language families and data conditions (Table 6.1): Romanian-English (Ro-En) from WMT16 (Bojar et al., 2016), English-German (En-De) from WMT14 (Bojar et al., 2014), and English-Japanese (En-Ja) from WAT2017 Small-NMT Task (Nakazawa et al., 2017). We also evaluate EDITOR on the two En-De test sets with terminology constraints released by Dinu et al. (2019). The test sets are subsets of the WMT17 En-De test set (Bojar et al., 2017) with terminology constraints extracted from Wiktionary and IATE.[6] For each test set, they only select the sentence pairs in which the exact target terms are used in the reference. The resulting Wiktionary and IATE test sets contain 727 and 414 sentences respectively. We follow the same preprocessing steps in Gu et al. (2019): we apply normalization, tokenization, true-casing, and BPE (Sennrich et al., 2016c) with 37k and 40k operations for En-De and Ro-En. For En-Ja, we use the provided subword vocabularies (16,384 BPE per language from SentencePiece (Kudo and Richardson, 2018)).

Experimental Conditions We train and evaluate the following models in controlled conditions to thoroughly evaluate EDITOR:

- **Auto-Regressive Transformers (AR)** built using Sockeye (Hieber et al., 2017) and fairseq (Ott et al., 2019). We report AR baselines with both toolkits to enable fair

[6] Available at `https://www.wiktionary.org/` and `https://iate.europa.eu`.

comparisons when using our fairseq-based implementation of EDITOR and Sockeye-based implementation of lexically constrained decoding algorithms (Post and Vilar, 2018).

- **Non Auto-Regressive Transformers (NAR)** In addition to **EDITOR**, we train a Levenshtein Transformer (**LevT**) with approximately the same number of parameters. Both are implemented using fairseq.

Model & Training Configurations All models adopt the *base* Transformer architecture (Vaswani et al., 2017) with $d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $n_{\text{heads}} = 8$, $n_{\text{layers}} = 6$, and $p_{\text{dropout}} = 0.3$. For En-De and Ro-En, the source and target embeddings are tied with the output layer weights (Nguyen and Chiang, 2018; Press and Wolf, 2017). We add dropout to embeddings (0.1) and label smoothing (0.1). AR models are trained with the Adam optimizer (Kingma and Ba, 2015) with a batch size of 4096 tokens. We checkpoint models every 1000 updates. The initial learning rate is 0.0002, and it is reduced by 30% after 4 checkpoints without validation perplexity improvement. Training stops after 20 checkpoints without improvement. All NAR models are trained using Adam (Kingma and Ba, 2015) with initial learning rate of 0.0005 and a batch size of 64,800 tokens for maximum 300,000 steps.[7] We select the best checkpoint based on validation BLEU (Papineni et al., 2002). All models are trained on 8 NVIDIA V100 Tensor Core GPUs.

Knowledge Distillation We apply sequence-level knowledge distillation from autoregressive teacher models as widely used in non-autoregressive generation (Gu et al., 2018, 2019; Lee et al., 2018a). Specifically, when training the non-autoregressive models, we replace the reference sequences $\boldsymbol{y}^*$ in the training data with translation outputs from the AR teacher

[7] Our preliminary experiments and prior work show that NAR models require larger training batches than AR models.

model (Sockeye, with beam = 4).[8] We also report the results when applying knowledge distillation to autoregressive models.

Evaluation We evaluate translation quality via case-sensitive tokenized **BLEU** (as in Gu et al. (2019))[9] and **RIBES** (Isozaki et al., 2010), which is more sensitive to word order differences. Before computing the scores, we tokenize the German and English outputs using Moses and Japanese outputs using KyTea.[10] For lexically constrained decoding, we report the constraint preservation rate (**CPR**) in the translation outputs.

We quantify decoding speed using **latency** per sentence. It is computed as the average time (in ms) required to translate the test set using batch size of one (excluding the model loading time) divided by the number of sentences in the test set.

6.3.2 MT Tasks

Since our experiments involve two different toolkits, we first compare the same Transformer AR models built with Sockeye and with fairseq: the AR models achieve comparable decoding speed and translation quality regardless of toolkit – the Sockeye model obtains higher BLEU than the fairseq model on Ro-En and En-De but lower on En-Ja (Table 6.2). Further comparisons will therefore center on the Sockeye AR model to better compare EDITOR with the lexically constrained decoding algorithm (Post and Vilar, 2018).

Table 6.2 also shows that knowledge distillation has a small and inconsistent impact on AR models (Sockeye): it yields higher BLEU on Ro-En, close BLEU on En-De, and lower BLEU on

[8] This teacher model was selected for a fairer comparison on MT with lexical constraints.

[9] https://github.com/pytorch/fairseq/blob/master/fairseq/clib/libbleu/libbleu.cpp

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

		Distill	Beam	Params	BLEU ↑	RIBES ↑	Latency (ms) ↓
Ro-En	AR (fairseq)		4	64.5M	32.0	83.8	357.14
	AR (sockeye)		4	64.5M	32.3	83.6	369.82
	AR (sockeye)		10	64.5M	32.5	83.8	394.52
	AR (sockeye)	✓	10	64.5M	32.9	84.2	371.75
	NAR: LevT	✓	–	90.9M	31.6	**84.0**	98.81
	NAR: EDITOR	✓	–	90.9M	**31.9**	**84.0**	**93.20**
En-De	AR (fairseq)		4	64.9M	27.1	80.4	363.64
	AR (sockeye)		4	64.9M	27.3	80.2	308.64
	AR (sockeye)		10	64.9M	27.4	80.3	332.73
	AR (sockeye)	✓	10	64.9M	27.6	80.5	363.52
	NAR: LevT	✓	–	91.1M	**26.9**	**81.0**	113.12
	NAR: EDITOR	✓	–	91.1M	**26.9**	80.9	**105.37**
En-Ja	AR (fairseq)		4	62.4M	44.9	85.7	292.40
	AR (sockeye)		4	62.4M	43.4	85.1	286.83
	AR (sockeye)		10	62.4M	43.5	85.3	311.38
	AR (sockeye)	✓	10	62.4M	42.7	85.1	295.32
	NAR: LevT	✓	–	106.1M	**42.4**	84.5	143.88
	NAR: EDITOR	✓	–	106.1M	42.3	**85.1**	**96.62**

Table 6.2: Machine Translation Results. For each metric, we underline the top scores among all models and boldface the top scores among NAR models. EDITOR decodes 6–7% faster than LevT on Ro-En and En-De, and 33% faster on En-Ja, while achieving comparable or higher BLEU and RIBES.

En-Ja.[11] Thus, we use the AR models trained without distillation in further experiments.

Next, we compare the NAR models against the AR (Sockeye) baseline. As expected, both EDITOR and LevT achieve close translation quality to their AR teachers with 2–4 times speedup. BLEU differences are small ($\Delta < 1.1$) as in prior work (Gu et al., 2019). The RIBES trends are more surprising: both NAR models significantly outperform the AR models (Sockeye) on RIBES, except for En-Ja, where EDITOR and the AR models significantly outperforms LevT.[12] This illustrates the strength of EDITOR in word reordering.

Finally, results confirm the benefits of EDITOR's reposition operation over LevT: decoding with EDITOR is 6–7% faster than LevT on Ro-En and En-De, and 33% faster on En-Ja – a more

[11] Kasai et al. (2021) found that AR models can benefit from knowledge distillation but with a Transformer large model as a teacher, while we use the Transformer base model.

[12] All mentions of significance on standard MT tasks are based on the paired bootstrap test with $p < 0.05$ (Clark et al., 2011).

		Distill	Beam	BLEU ↑	RIBES ↑	CPR ↑	Latency (ms) ↓
Ro-En	AR + DBA (sockeye)		4	31.0	79.5	99.7	436.26
	AR + DBA (sockeye)		10	34.6	84.5	99.5	696.68
	NAR: LevT	✓	–	31.6	83.4	80.3	121.80
	+ hard constraints	✓	–	27.7	78.4	**99.9**	140.79
	NAR: EDITOR	✓	–	**33.1**	**85.0**	86.8	**108.98**
	+ hard constraints	✓	–	28.8	81.2	95.0	136.78
En-De	AR + DBA (sockeye)		4	26.1	74.7	99.7	434.41
	AR + DBA (sockeye)		10	30.5	81.9	99.5	896.60
	NAR: LevT	✓	–	27.1	80.0	75.6	127.00
	+ hard constraints	✓	–	24.9	74.1	**100.0**	134.10
	NAR: EDITOR	✓	–	**28.2**	**81.6**	88.4	**121.65**
	+ hard constraints	✓	–	25.8	77.2	96.8	134.10
En-Ja	AR + DBA (sockeye)		4	44.3	81.6	100.0	418.71
	AR + DBA (sockeye)		10	48.0	85.9	100.0	736.92
	NAR: LevT	✓	–	42.8	84.0	74.3	161.17
	+ hard constraints	✓	–	39.7	77.4	**99.9**	159.27
	NAR: EDITOR	✓	–	**45.3**	**85.7**	91.3	**109.50**
	+ hard constraints	✓	–	43.7	82.6	96.4	132.71

Table 6.3: Machine Translation with lexical constraints (averages over 5 runs). For each metric, we underline the top scores among all models and boldface the top scores among NAR models. EDITOR exploits constraints better than LevT. It also achieves comparable RIBES to the best AR model with 6–7 times decoding speedup.

distant language pair which requires more reordering but no inflection changes on reordered words – with no statistically significant difference in BLEU nor RIBES, except for En-Ja, where EDITOR significantly outperforms LevT on RIBES. Overall, EDITOR is shown to be a good alternative to LevT on standard machine translation tasks and can also be used to replace the AR models in settings where decoding speed matters more than small differences in translation quality.

6.3.3 MT with Lexical Constraints

We now turn to the main evaluation of EDITOR on machine translation with lexical constraints.

Experimental Conditions We conduct a controlled comparison of the following approaches:

- NAR models: **EDITOR** and **LevT** view the lexical constraints as **soft constraints**, provided via the initial target sequence. We also explore the decoding technique introduced in Susanto et al. (2020) to support **hard constraints**.
- AR models: they use the provided target words as hard constraints enforced at decoding time by an efficient form of constrained beam search: dynamic beam allocation (**DBA**) (Post and Vilar, 2018).[13]

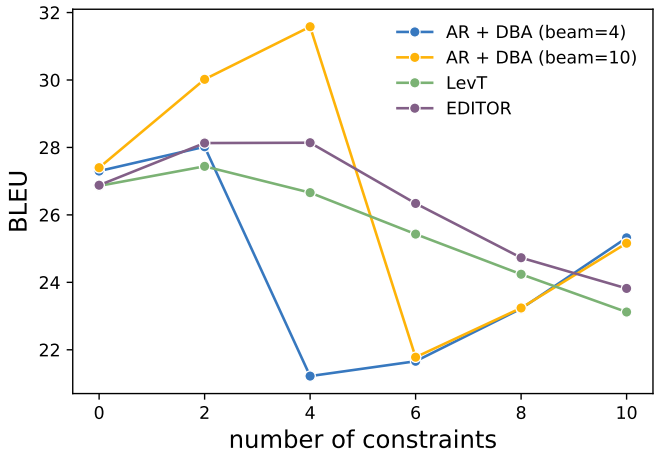
Crucially, all models, including EDITOR, are the exact same models evaluated on the standard MT tasks above, and do not need to be trained specifically to incorporate constraints.

We define lexical constraints as Post and Vilar (2018): for each source sentence, we randomly select one to four words from the reference as lexical constraints. We then randomly shuffle the constraints and apply BPE to the constraint sequence. Different from the terminology test sets in Dinu et al. (2019) which contain only several hundred sentences with mostly nominal constraints, our constructed test sets are larger and include lexical constraints of all types.

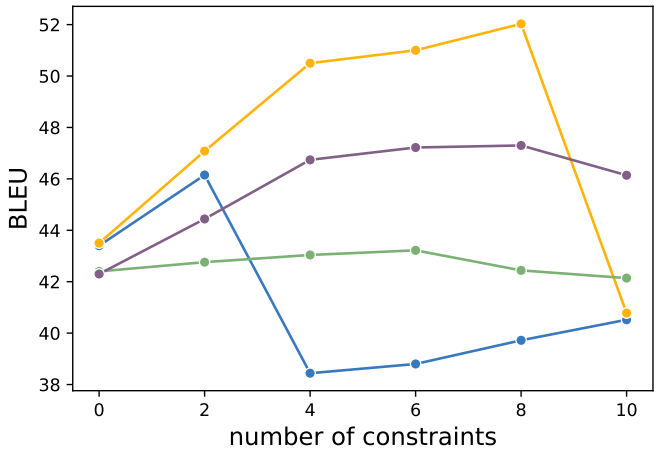
Main Results Table 6.3 shows that EDITOR exploits the soft constraints to strike a better balance between translation quality and decoding speed than other models. Compared to LevT, EDITOR preserves 7–17% more constraints and achieves significantly higher translation quality (+1.1–2.5 on BLEU and +1.6–1.8 on RIBES) and faster decoding speed.[14] Compared to the AR model with beam = 4, EDITOR yields significantly higher BLEU (+1.0–2.2) and RIBES (+4.1–6.9) with 3–4 times decoding speedup. After increasing the beam to 10, EDITOR obtains lower BLEU

[13] Although the beam pruning option in Post and Vilar (2018) is not used here (since it is not supported in Sockeye anymore), other Sockeye updates improve efficiency. Constrained decoding with DBA is 1.8–2.7 times slower than unconstrained decoding here, while DBA is 3 times slower when beam = 10 in Post and Vilar (2018).

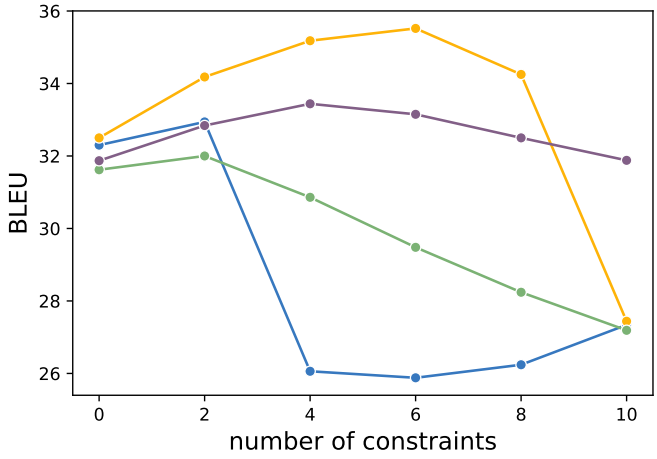
[14] All mentions of significance on the lexically constrained MT tasks are based on the independent student's t-test with Bonferroni Correction ($p < 0.05$) over 5 runs.



(a) En-De

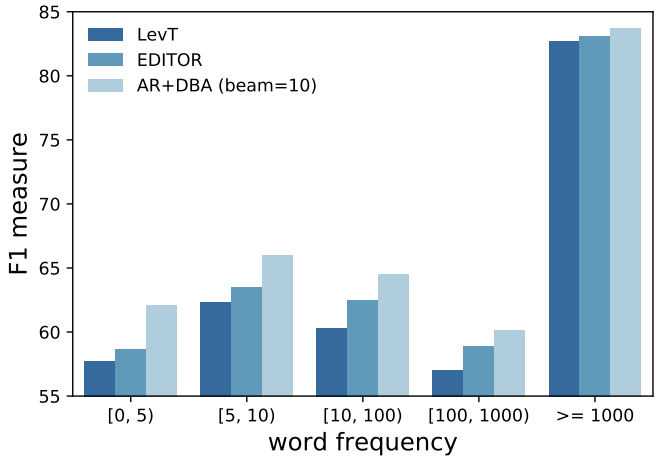


(b) En-Ja

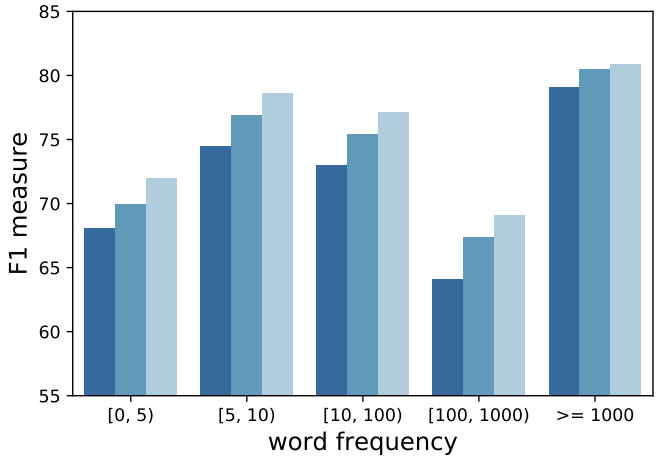


(c) Ro-En

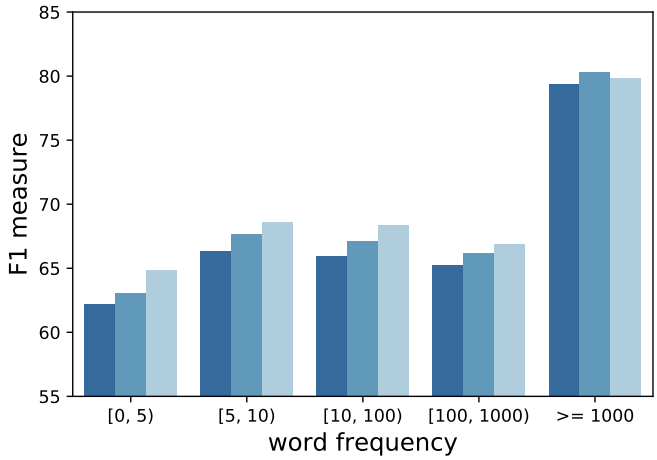
Figure 6.5: EDITOR improves BLEU over LevT for 2–10 constraints (counted pre-BPE) and beats the best AR model on 2/3 tasks with 10 constraints.



(a) En-De



(b) En-Ja



(c) Ro-En

Figure 6.6: Target word F1 score binned by word test set frequency: EDITOR improves over LevT the most for words of low or medium frequency. AR achieves higher F1 than EDITOR for words of low or medium frequency at the cost of much longer decoding time.

but comparable RIBES with 6–7 times decoding speedup.[15] Note that AR models treat provided words as hard constraints and therefore achieve over 99% CPR by design, while NAR models treat them as soft constraints.

Results confirm that enforcing hard constraints increases CPR but degrades translation quality compared to the same model using soft constraints: for LevT, it degrades BLEU by 2.2–3.9 and RIBES by 5.0–6.6. For EDITOR, it degrades BLEU by 1.6–4.3 and RIBES by 3.1–4.4 (Table 6.3). By contrast, EDITOR with soft constraints strikes a better balance between translation quality and constraint preservation.

The strengths of EDITOR hold when varying the number of constraints (Figure 6.5). For all tasks and models, adding constraints helps BLEU up to a certain point, ranging from 4 to 10 words. When excluding the slower AR model (beam = 10), EDITOR consistently reaches the highest BLEU score with 2–10 constraints: EDITOR outperforms LevT and the AR model with beam = 4. Consistent with Post and Vilar (2018), as the number of constraints increases, the AR model needs larger beams to reach good performance. When the number of constraints increases to 10, EDITOR yields higher BLEU than the AR model on En-Ja and Ro-En, even after incurring the cost of increasing the AR beam to 10.

Are EDITOR improvements limited to preserving constraints better? We verify that this is not the case by computing the target word F1 binned by frequency (Neubig et al., 2019). Figure 6.6 shows that EDITOR improves over LevT across all test frequency classes and closes the gap between NAR and AR models: the largest improvements are obtained for low and medium frequency words – on En-De and En-Ja, the largest improvements are on words with frequency

[15] Post and Vilar (2018) show that the optimal beam size for DBA is 20. Our experiment on En-De shows that increasing the beam size from 10 to 20 improves BLEU by 0.7 at the cost of doubling the decoding time.

between 5 and 1000, while on Ro-En, EDITOR improves more on words with frequency between 5 and 100. EDITOR also improves F1 on rare words (frequency in $[0,5)$), but not as much as for more frequent words.

We now conduct further analysis to better understand the factors that contribute to EDITOR's advantages over LevT.

Impact of Reposition We compare the average number of basic edit operations (Section 6.2.1) of different types used by EDITOR and LevT on each test sentence (averaged over the 5 runs): reposition (excluding deletion for controlled comparison with LevT), deletion, and insertion performed by LevT and EDITOR at decoding time. Table 6.4 shows that LevT deletes tokens 2–3 times more often than EDITOR, which explains its lower CPR than EDITOR. LevT also inserts tokens 1.2–1.6 times more often than EDITOR and performs 1.4 times more edit operations on En-De and En-Ja. On Ro-En, LevT performs -4% fewer edit operations in total than EDITOR but is overall slower than EDITOR, since multiple operations can be done in parallel at each action step. Overall, EDITOR takes 3–40% fewer decoding iterations than LevT. These results suggest that reposition successfully reduces redundancy in edit operations and makes decoding more efficient by replacing sequences of insertions and deletions with a single repositioning step.

Furthermore, Figure 6.7 illustrates how reposition increases flexibility in exploiting lexical constraints, even when they are provided in the wrong order. While LevT generates an incorrect output by using constraints in the provided order, EDITOR's reposition operation helps generate a more fluent and adequate translation.

	Repos.	Del.	Ins.	Total	Iter.
Ro-En					
LevT	0.00	4.61	33.05	37.67	2.01
EDITOR	8.13	2.50	28.68	39.31	1.81
En-De					
LevT	0.00	7.13	45.45	52.58	2.14
EDITOR	5.85	4.01	28.75	38.61	2.07
En-Ja					
LevT	0.00	5.24	32.83	38.07	2.93
EDITOR	4.73	1.69	21.64	28.06	1.76

Table 6.4: Average number of repositions (excluding deletions), deletions, insertions, and decoding iterations to translate each sentence with soft lexical constraints (averaged over 5 runs). Thanks to reposition operations, EDITOR uses 40–70% fewer deletions, 10–40% fewer insertions, and 3–40% fewer decoding iterations overall.

Source:		Cred că Stephen Thompson are încredere în noi .
Reference:		I think Stephen Thompson has faith in us .
Constraints:		faith Stephen think
LevT:		
	y^0:	faith Stephen think
	$y' = \mathcal{E}(y^0, d^1)$:	faith Stephen think
Action a^1:	$y'' = \mathcal{E}(y', p^1)$:	[plh] [plh] faith [plh] Stephen [plh] [plh] [plh] [plh] think [plh]
	$y^1 = \mathcal{E}(y'', t^1)$:	I think faith that Stephen Thom@@ p@@ son can think .
	no further actions:	**[Terminate]**
EDITOR:		
	y^0:	faith Stephen think
	$y' = \mathcal{E}(y^0, r^1)$:	think Stephen faith
Action a^1:	$y'' = \mathcal{E}(y', p^1)$:	[plh] think Stephen [plh] [plh] [plh] [plh] faith [plh] [plh] [plh]
	$y^1 = \mathcal{E}(y'', t^1)$:	I think Stephen Thom@@ p@@ son has faith in us .
	no further actions:	**[Terminate]**

Figure 6.7: Ro-En translation with soft lexical constraints: while LevT uses the constraints in the provided order, EDITOR's reposition operation helps generate a more fluent and adequate translation.

Impact of Dual-Path Roll-In Ablation experiments (Table 6.5) show that EDITOR benefits greatly from dual-path roll-in. Replacing dual-path roll-in with the simpler roll-in policy used in Gu et al. (2019), the model's translation quality drops significantly (by 0.9–1.3 on BLEU and 0.6–1.9 on RIBES) with fewer constraints preserved and slower decoding. It still achieves better translation quality than LevT thanks to the reposition operation: specifically, it yields significantly higher BLEU and RIBES on Ro-En, comparable BLEU and significantly higher RIBES

	BLEU ↑	RIBES ↑	CPR↑	Lat.↓
Ro-En				
EDITOR	33.1	85.0	86.8	108.98
-dual-path	32.2	84.4	74.8	119.61
LevT	31.6	83.4	80.3	121.80
En-De				
EDITOR	28.2	81.6	88.4	121.65
-dual-path	27.2	80.4	78.7	130.85
LevT	27.1	80.0	75.6	127.00
En-Ja				
EDITOR	45.3	85.7	91.3	109.50
-dual-path	44.0	83.9	80.0	154.10
LevT	42.8	84.0	74.3	161.17

Table 6.5: Ablating the dual-path roll-in policy hurts EDITOR on soft-constrained MT, but still outperforms LevT, confirming that reposition and dual-path imitation learning both benefit EDITOR.

on En-De, and comparable RIBES and significantly higher BLEU on En-Ja than LevT.

6.3.4 MT with Terminology Constraints

We evaluate EDITOR on the terminology test sets released by Dinu et al. (2019) to test its ability to incorporate terminology constraints and to further compare it with prior work (Dinu et al., 2019; Post and Vilar, 2018; Susanto et al., 2020).

Compared to Post and Vilar (2018) and Dinu et al. (2019), EDITOR with soft constraints achieves higher absolute BLEU, and higher BLEU improvements over its counterpart without constraints (Table 6.6). Consistent with previous findings by Susanto et al. (2020), incorporating soft constraints in LevT improves BLEU by +0.3 on Wiktionary and by +0.4 on IATE. Enforcing hard constraints as in Susanto et al. (2020) increases the term usage by +8–10% and improves BLEU by +0.3–0.6 over LevT using soft constraints.[16] For EDITOR, adding soft constraints improves BLEU by +0.5 on Wiktionary and +0.9 on IATE, with very high term usages (96.8% and

[16]We use our implementations of Susanto et al. (2020)'s technique for a more controlled comparison. The LevT baseline in Susanto et al. (2020) achieves higher BLEU than ours on the small Wiktionary and IATE test sets, while it underperforms our LevT on the full WMT14 test set (26.5 vs. 26.9).

	Wiktionary		IATE	
	Term%↑	BLEU ↑	Term%↑	BLEU ↑
Prior Results				
Base Trans.	76.9	26.0	76.3	25.8
Post18	99.5	25.8	82.0	25.3
Dinu19	93.4	26.3	94.5	26.0
Base LevT	81.1	30.2	80.3	29.0
Susanto20	100.0	31.2	100.0	30.1
Our Results				
LevT	84.3	28.2	83.9	27.9
+ soft constraints	90.5	28.5	92.5	28.3
+ hard constraints	100.0	28.8	100.0	28.9
EDITOR	83.5	28.8	83.0	27.9
+ soft constraints	96.8	29.3	97.1	28.8
+ hard constraints	99.8	29.3	100.0	28.9

Table 6.6: Term usage percentage (*Term%*) and BLEU scores of En-De models on terminology test sets (Dinu et al., 2019) provided with correct terminology entries (exact matches on both source and target sides). EDITOR with soft constraints achieves higher BLEU than LevT with soft constraints, and on par or higher BLEU than LevT with hard constraints.

97.1% respectively). EDITOR thus correctly uses the provided terms almost all the time when they are provided as soft constraints, so there is little benefit to enforcing hard constraints instead: they help close the small gap to reach 100% term usage and do not improve BLEU. Overall, EDITOR achieves on par or higher BLEU than LevT with hard constraints.

Results also suggest that EDITOR can handle phrasal constraints even though it relies on token-level edit operations, since it achieves above 99% term usage on the terminology test sets where 26–27% of the constraints are multi-token.

6.3.5 Limitations

The findings of this work should be interpreted with several limitations in mind. First, we evaluate EDITOR on three translation tasks ranging from medium to high-resource settings, while it remains an open question how EDITOR would perform in other settings, especially in the truly low-resource settings. Second, we compare the decoding speed of AR and NAR models based

on latency (i.e. decoding with a batch size of 1 on a single GPU), while Helcl et al. (2022); Kasai et al. (2021) point out that AR models can achieve competitive or even faster decoding speed than NAR models when using larger batch sizes on standard MT tasks. Thus, further experiments are needed to systematically compare the decoding speed of these models under various batching and hardware setups on lexically constrained MT. Finally, in the experiments of lexically constrained MT, the constraints are automatically extracted and matched with the reference, while in real applications, the constraints are typically provided by users or extracted from dictionaries. Therefore, it needs to be further investigated how EDITOR performs in more realistic settings and how humans perceive the outputs and affordances of EDITOR.

6.4 Summary

We introduced EDITOR, a non-autoregressive transformer model that iteratively edits hypotheses using a novel reposition operation. Reposition combined with a new dual-path imitation learning strategy helps EDITOR generate output sequences that can be flexibly controlled by user's lexical choice preferences. Extensive experiments show that EDITOR exploits soft lexical constraints more effectively than the Levenshtein Transformer (Gu et al., 2019) while speeding up decoding dramatically compared to constrained beam search (Post and Vilar, 2018). Results also confirm the benefits of using soft constraints over hard ones in terms of translation quality. EDITOR also achieves comparable or better translation quality with faster decoding speed than the Levenshtein Transformer on three standard MT tasks.

This work also leaves open questions. For example, can we incorporate users' preferences or constraints in more flexible forms? For example, it would be easier for users to provide

constraints in dictionary forms without having to predict their inflection forms based on the sentence-level context. We aim to answer this question in the next chapter by introducing a modular framework for inflecting and incorporating the lemma-form constraints in NMT.

Chapter 7: Stronger Inductive Biases for Flexible Constraint Integration

7.1 Introduction

In the previous chapter, we mainly focus on incorporating constraint terms in the output exactly as given. This is problematic when translating into morphologically rich languages where constraints need be adequately inflected in the output, while it is more natural and flexible to provide constraints as lemmas as in a dictionary. Thus in this chapter, we aim to design a framework that can automatically inflect and incorporate lemma-form constraints in NMT outputs.

To the best of our knowledge, only one paper has directly addressed this problem for neural MT: (Bergmanis and Pinnis, 2021) design an NMT model trained to copy-and-inflect the constraint words using target lemma annotations (TLA) — TLA are synthetic training samples where the source sentence is tagged with automatically generated lemma constraints. While this approach improves translation quality, the end-to-end training set-up prevents fast adaptation to lemmas and inflected forms that are rare or unseen at training time. Its impact is also limited to a specific neural architecture, and it is unclear whether its benefits port to more generic sequence-to-sequence models.

In this chapter, we introduce a modular framework for inflecting and integrating constraint terms in NMT. It relies on a cross-lingual inflection module that predicts the inflected form of each lemma constraint based on the source context only. The inflected lemmas can then

be incorporated into NMT using any of the aforementioned constrained NMT techniques. Compared with TLA, this framework is more flexible, as it can be applied to diverse types of NMT architectures and inflection modules, and facilitates fast adaptation to new terms without retraining the base NMT model from scratch. This flexibility is enabled by the cross-lingual nature of the inflection module, which predicts the inflected form of each target lemma based on the source context only. This differs from traditional inflection models that predict the inflected forms based on pre-specified morphological tags or monolingual target context.

Based on this framework, we make the following contributions:

- We construct and release test suites to evaluate models' ability to inflect constraint terms for domain adaptation (English-German Health) and low-resource MT (English-Lithuanian News).
- We show that integrating linguistic knowledge through a simple rule-based inflection module improves over its neural counterpart in intrinsic and end-to-end MT evaluations.
- Our framework improves autoregressive and non-autoregressive translation, and outperforms the existing TLA approach for inflecting lemma constraints. We open-source the code to facilitate replication and extensions.

7.2 Inflecting Target Lemmas Given the Source Context

We introduce a modular framework for inflecting terminology constraints for NMT, where we first build an inflection module that predicts the inflected form of each target lemma term based on the source sentence and then incorporate the inflected constraints in NMT using any of the aforementioned techniques. By framing the problem this way, we assume that the inflected

forms can be inferred based only on the source context and integrated in a fluent translation by NMT models. In cases where there are multiple possible inflected forms corresponding to different ways of translating the source, the inflection module can predict one of the possible forms, and the NMT model can generate a translation conditioned on the predicted forms of the constraints. Compared with Bergmanis and Pinnis (2021), our framework is more flexible – it can be combined with any NMT model that enables translation with constraints and can leverage diverse types of morphological inflection modules in which linguistic knowledge can be easily incorporated.

Formally, given a source sequence $\boldsymbol{x}$ and k target lemma words $\bar{\boldsymbol{z}} = (\bar{z}_1, \bar{z}_2, ..., \bar{z}_k)$ that need to be inflected, the inflection module Θ predicts the inflected form of each target lemma $\boldsymbol{z} = (z_1, z_2, ..., z_k)$ independently:

$$p(\boldsymbol{z} \,|\, \boldsymbol{x}, \bar{\boldsymbol{z}}; \Theta) = \prod_{i=1}^{k} p(z_i \,|\, \boldsymbol{x}, \bar{z}_i; \Theta) \tag{7.1}$$

7.2.1 Rule-Based Inflection Module

One can predict the inflected form of a target word given its lemma and the source context in two steps: first predict the morphological tag of the target word based on the source context, and then predict the inflected form based on the lemma and morphological tag. The second step can be modeled using traditional inflection models (Cotterell et al., 2017), while the first step can be performed using rule-based inference based on linguistic knowledge. McCarthy et al. (2020) present a universal morphological (UniMorph) paradigm with universal morphological tags for hundreds of world languages. In UniMorph, the morphological tag of a verb includes

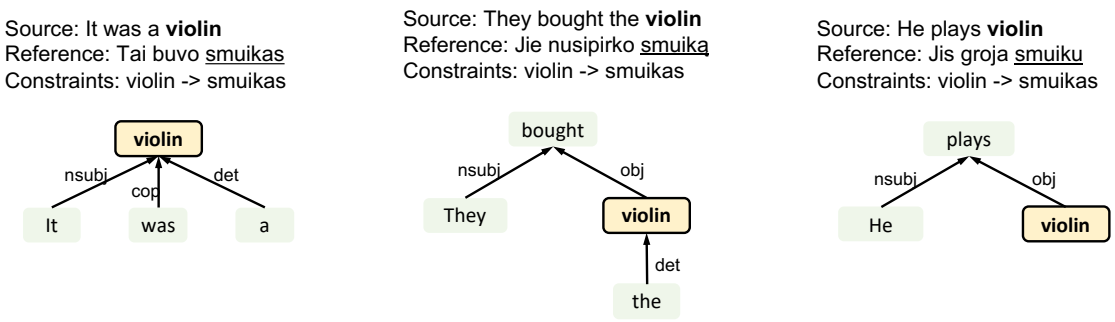
information about the tense (past, present, or future), mood (indicative, conditional, imperative, or subjunctive), the number (singular or plural) and person (first, second, or third person) of the subject. The tag of a noun or adjective includes information about gender (masculine or feminine), number, and grammatical case. Some of these can be inferred from the target lemma (e.g. the gender of a noun) or the source term (e.g. the number of a noun), while some others need to be inferred based on the grammatical function of the source term in the sentence (e.g. grammatical case) or the sentence-level semantics (e.g. mood). Many of the inference rules are shared across a wide range of languages, except for the tense and mood of verbs, as well as the gender and some grammatical cases of nouns and adjectives.

In our rule-based inflection module, we extract the morphological features, part-of-speech tags, and dependency parsing tree of the source sentence using pre-trained Stanza models[1] and infer the aforementioned classes based on grammar rules and validation examples. The tense and mood of a verb are inferred from the morphological form of the corresponding source term,[2] while the number and person of its subject are inferred based on the morphological form of its subject. For nouns and adjectives, the number can be inferred from the morphological form of the source term or modified noun, while the gender can be determined based on the target lemma.

To infer the grammatical case of a noun or adjective, one needs to infer about the grammatical role of the source term in the sentence. For example, in Lithuanian, there are seven main cases, including nominative, genitive, dative, accusative, instrumental, locative, and vocative cases. Figure 7.1 shows examples of how the case of a Lithuanian noun can be inferred from the dependency parsing tree of the source sentence. Some of the cases can be

[1] `https://github.com/stanfordnlp/stanza`

[2] We ignore tense and mood types that cannot be inferred from the source term.



(a) Nominative Case (b) Accusative Case (c) Instrumental Case

Figure 7.1: Examples showing how the grammatical case of a target lemma is inferred from the dependency parsing tree of the source sentence. In each example, the reference usage of the target constraint is underlined, and its corresponding source term is boldfaced and highlighted in the yellow, outlined box in each dependency tree. Figure (a) shows an example where the constraint term "smuikas" is used in *nominative* case in the reference, since it is the root in the dependency tree. In Figure (b), the same constraint term is used in *accusative* case in the reference, since it is the *object* of the root verb "bought". However, not all objects should be used in *accusative* case. As shown in Figure (c), "smuikas" is used in *instrumental* case, since it serves as the *instrument* with which the subject performs the action.

easily distinguished from the others, while some are more difficult to infer. In this example, the nominative case is comparatively easy to infer – the noun should be in the nominative case when the corresponding source term is the root or subject of the sentence. However, to distinguish between dative, accusative, instrumental, and locative cases, one needs to infer based on the grammatical and semantic role of the source term. In our rule-based module, we only take into account the most common scenarios.[3]

Finally, given a lemma and its morphological tag, one can look up its inflected form in a morphological dictionary. We use DEMorphy (Altinok, 2018) for German and Wiktionary[4] for Lithuanian. Since most Lithuanian nouns follow a set of declension rules,[5] we inflect Lithuanian nouns based on the rules for lemmas unseen in the dictionary.

[3]Our code only includes a few simple inference rules written by non-expert based on the grammar knowledge from Wikipedia pages.

[4]https://www.wiktionary.org/

[5]https://en.wikipedia.org/wiki/Lithuanian_declension

7.2.2 Neural Inflection Module

As prior work shows that BERT-style architectures (Devlin et al., 2019) can encode morphological information in their hidden representations and disambiguate morphologically ambiguous forms via contextualized encoding (Edmiston, 2020), we build the neural-based inflection module as a substitution model and base it on the encoder-decoder Transformer architecture, which embeds the source sentence through the encoder and the target lemmas through the decoder. Next, the decoder predicts the inflected form of each target word in parallel. The inflection module resembles the architecture of the conditional masked language model (CMLM) (Ghazvininejad et al., 2019) but differs in decoder input and output: CMLM takes the target sentence with some tokens masked out as input and is trained to predict only the masked tokens conditioned on unmasked ones, while our inflection module takes target tokens in their lemma forms as input and predicts their inflected forms.

CMLM only allows for one-to-one substitution of subwords. However, in the case of inflection, the number of subwords that constitute a lemma and its inflected form may differ. To facilitate varying-length substitution, we construct the decoder input by inserting K placeholders at the end of each target lemma. Next, the model predicts the token $t \in \mathcal{V} \cup \{\text{[PLH]}\}$ to be inserted at each input position. If $t = \text{[PLH]}$, we delete the token at this position, otherwise we replace the token at this position with t.[6]

[6] So for instance, given the input "freeze [PLH] [PLH]", the model could predict the output "fro@@ zen [PLH]".

7.3 Evaluation Test Suites

To evaluate the models' ability to incorporate diverse types of lemma constraints in different context, we choose the two morphologically complex languages – German and Lithuanian – as the target languages, both of which are fusional languages with strong suffixing. We create two test suites – the English→German health test suite[7] to evaluate models in the domain adaptation scenario and English→Lithuanian news test suite to test models in the low-resource setting. Different from the automotive test suite of Bergmanis and Pinnis (2021), which contains short sentences (15 tokens per source sentence on average) annotated with limited types of constraints (mostly nouns and proper nouns), our test suites contains longer sentences (20 and 25 tokens per source sentence on average) and diverse types of constraints including adjectives, nouns, proper nouns, and verbs. Different from the upcoming WMT21 terminology task[8] where the terminology translation table includes different forms for a given source term, our test suites only provides terminology translations in lemma forms.

Health Test Suite We construct the health test suite to test the models' ability to integrate terminology translations for fast domain adaptation. The test set contains English health information text annotated with domain-specific terminology translations and the human-translated sentences in German. We extract English→German test examples from the Himl Test Set,[9] which consists of English health information texts manually translated into German. We extract keyphrases from each source sentence using Yet Another Keyword

[7]To the best of our knowledge, there is no public health (or any non-news) domain MT test set for English→Lithuanian.

[8]http://statmt.org/wmt21/terminology-task.html

[9]http://www.himl.eu/test-sets

	Source	Constraints	Reference
En-De Health	The routine use of **abdominal** drainage to reduce postoperative complications after **appendectomy** for complicated **appendicitis** is controversial.	**abdominal** abdominell **appendectomy** Appendektomie **appendicitis** Appendizitis	Die routinemäßige Verwendung von *abdomineller* Drainage zur Verminderung postoperativer Komplikationen nach einer Appendektomie bei komplizierter Appendizitis ist umstritten.
En-Lt News	A fire in 1939 left the building badly damaged, but as Father **Johnson**'s parishioners made plans to rebuild , they commissioned the **carillon**.	**Johnson** Johnsonas **carillon** karilionas	1939 m. kilęs gaisras smarkiai apgadino pastatą, tačiau Tėvo *Johnsono* parapijiečiai planavo jį atstatyti, todėl užsakė *karilioną*.
	The expert who played the **carillon** in July called it something else: "A cultural treasure" and "an irreplaceable historical instrument".	**carillon** karilionas	Liepos mėnesį *karilionu* grojęs ekspertas pavadino jį kitaip: "kultūros lobiu" ir "nepakeičiamu istoriniu instrumentu".

Table 7.1: Examples from the English→German (En-De) health and English→Lithuanian (En-Lt) news test suites. For En-Lt, we select two examples from the same document. The annotated source terms are boldfaced and the target constraint terms are underlined. Some terms can be copied to the target (e.g. "Lymphödem" and "klinisch" in En-De), while some others need to be inflected in the target sentence (italicized).

	#Sent	#Const	#Const.Inf
Health	3000	4589	802
News	823	374	132

Table 7.2: Number of sentences (*#Sent*), constraints (*#Const*), and constraints that need to be inflected (*#Const.Inf*) in the health and news test suites.

Extractor (YAKE) (Campos et al., 2020)[10] and filter out phrases with high or medium frequency in the training corpora since they are mostly common and domain-generic phrases.[11] We extract terminology translations from WikiTitles[12] and an online English-German dictionary,[13] and annotate the keyphrases whose dictionary translations match the reference translation. As shown in Table 7.1, each source sentence in the test set is annotated with health-related terminology translations in the lemma forms, some of which can be directly copied to the final translation

[10] YAKE extracts n-grams as keyphrases based on word casing, frequency, position, and their sentence context.

[11] We filter out keyphrases with frequency > 100 in the WMT news training data.

[12] http://data.statmt.org/wikititles/v1/ and http://data.statmt.org/wikititles/v2/

[13] https://www.dict.cc/

while some need to be inflected based on the context.

News Test Suite The news test suite simulates the scenario where a user looks up keyphrases of a document in a bilingual dictionary and pick the top translation for each keyphrase as a constraint to help low-resource MT. We choose English→Lithuanian as an example of low-resource translation. The test suite is constructed from English→Lithuanian test examples from WMT 2019 news test sets. We first extract keyphrases from each source document using YAKE. Then, we find the top translation of each keyphrase (for many terms there's only one translation available) in an online dictionary.[14] We filter out the keyphrases whose translations do not match the reference. Table 7.1 shows two examples from the same document in the test suite. All occurrences of a keyphrase in one document are annotated with its target translation to encourage consistent translation of keyphrases within a document.[15] Table 7.2 shows the number of sentences and constraints in each test suite.

7.4 Experiments

7.4.1 Experimental Settings

Training Data For English→German (En-De), we use the training corpora from WMT14 (Bojar et al., 2014) and *newstest2013* for validation. For English→Lithuanian, we use the training data from WMT19 (Barrault et al., 2019) and *newsdev2019* as the validation set. For preprocessing, we apply normalization, tokenization, true-casing, and BPE (Sennrich et al., 2016c) with 37,000

[14] `https://lithuanian.english-dictionary.help`

[15] Interestingly, in Lithuanian, the masculine foreign names are usually translated by appending a suffix to the name to reflect their inflection forms. In this example, the foreign name "Johnson" is translated into "Johnsonas" in the nominative form in the dictionary, while in the reference it becomes "Johnsono" in the genitive form.

and 24,500 merging operations for En-De and En-Lt. Table 7.3 shows the provenance and statistics of the preprocessed data.

	Train	Valid	Provenance
En-De	3,961k	3,000	WMT14
En-Lt	1,612k	1,964	WMT19

Table 7.3: Number of sentence pairs and provenance of the training and validation data.

Baselines We compare our model with the following baselines:

- **Auto-Regressive (AR) baseline** without integrating terminology constraints.
- **AR with Constrained Decoding (CD)** to incorporate hard constraints (Post, 2018).
- **AR with Target Lemma Annotation** (TLA) that integrates lemma constraints as an additional input stream on the source side (Bergmanis and Pinnis, 2021).
- **Non-AutoRegressive (NAR) baseline** based on the EDITOR model (Xu and Carpuat, 2021b).
- **NAR with constraints (NAR+C)** that integrates constraints as the initial sequence in EDITOR without explicit inflection.

MT Models All models are based on the *base* Transformer (Vaswani et al., 2017) with $d_{\text{model}} = 512$, $d_{\text{hidden}} = 2048$, $n_{\text{heads}} = 8$, $n_{\text{layers}} = 6$, and $p_{\text{dropout}} = 0.3$. We tie the source and target embeddings with the output layer weights (Nguyen and Chiang, 2018; Press and Wolf, 2017). We add dropout to embeddings (0.1) and label smoothing (0.1). All models are trained with the Adam optimizer (Kingma and Ba, 2015) with initial learning rate of 0.0005 and effective batch sizes of $32k$ tokens for AR models and $64k$ tokens for NAR models for maximum 300,000 steps.[16] We

[16] As shown in prior work, the batch sizes for training non-autoregressive models are typically larger than the AR model (Zhou et al., 2020).

select the best checkpoint based on validation perplexity. Following Xu and Carpuat (2021b), we train NAR models using sequence-level knowledge distillation: we replace the reference sentences in the training data with translation outputs from the AR models. For decoding, we use beam search with a beam size of 4 for AR and AR with TLA, while for AR with CD we use a beam size of 20 as suggested in prior work (Post and Vilar, 2018). To enhance constraint usage in NAR models, we adopt the techniques by Susanto et al. (2020): we prohibit deletions on constraint tokens or insertions within the constraint segments.

Neural Inflection Model Its synthetic training data is derived from the MT parallel data. We first lemmatise and part-of-speech tag the target sentences using Stanza. We then randomly select adjectives, verbs, nouns, and proper nouns from each target sentence and train the inflection module to predict their inflected forms based on their lemma forms and the source sentence. Following Bergmanis and Pinnis (2021), we draw the proportion of words selected in each target sentence randomly from the uniform distribution between $(0, 0.4]$. For training, we initialize its encoder parameters using the NAR baseline encoder and train it using Adam optimizer with a batch size of $32k$ tokens for maximum $200k$ steps.

All models are trained on 2 GeForce GTX 1080 Ti GPUs. Table 7.4 shows the number of parameters in each model.

Evaluation We evaluate translation quality using sacreBLEU (Post, 2018). To evaluate how well the translation preferences are incorporated in the translation outputs, we measure **lemma usage rate** by first lemmatising the translation output and then computing the percentage of lemma terms that appear in the lemmatised output. To evaluate whether the terms are inflected

	Model Size (M)
En-De	
AR	65
AR w/ CD	65
AR w/ TLA	65
NAR	91
rule-based inflection	0
neural-based inflection	86
En-Lt	
AR	57
AR w/ CD	57
AR w/ TLA	58
NAR	84
rule-based inflection	0
neural-based inflection	72

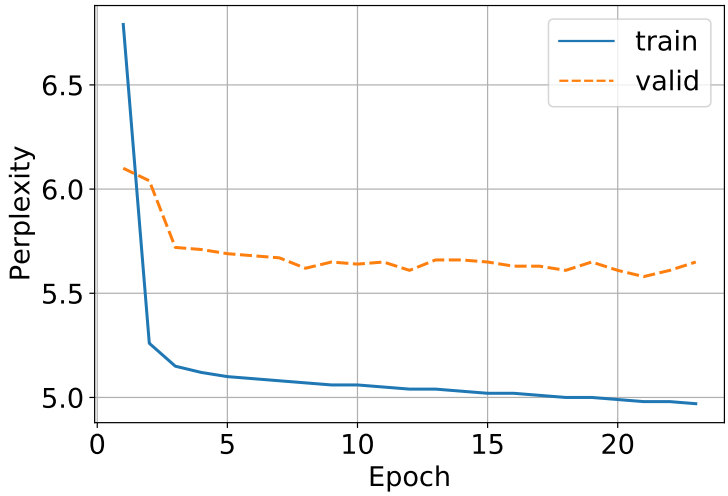
Table 7.4: Model sizes (M) for the AR, NAR, and inflection models.

correctly, we measure **term usage accuracy** by matching each lemma constraint with its inflected form in the reference and computing the percentage of reference inflected terms that appear in the translation output.

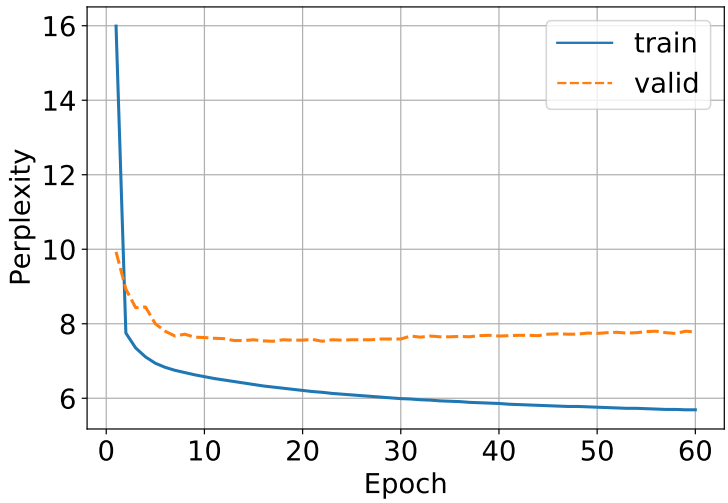
7.4.2 Results and Discussion

Intrinsic Inflection Accuracy To evaluate the quality of the inflection modules, we first compare the inflection accuracy of neural-based and rule-based inflection modules against the term usage accuracy of the TLA model. The rule-based inflection module achieves higher inflection accuracy than the neural-based module on both test suites: the neural-based module obtains 81.2% accuracy on En-De health set and 15.4% accuracy on En-Lt news set, while the rule-based module achieves 87.6% accuracy on En-De and 77.4% accuracy on En-Lt. The rule-based module achieves close accuracy to TLA on En-De (89.2% term usage accuracy) and higher accuracy on En-Lt (67.9% term usage accuracy).

To investigate why the neural-based inflection underperforms the rule-based one, we



(a) En-De



(b) En-Lt

Figure 7.2: Learning curves of the training and validation perplexity for the En-De and En-Lt neural-based inflection modules.

examine how the training and validation perplexity changes over the number of training epochs (Figure 7.2). On both languages, the validation perplexity stops decreasing after a few training epochs (10 epochs for En-De and 20 epochs for En-Lt) while the training perplexity decreases very slowly. The final training perplexity remains at around 5.1 on En-De and 5.7 on En-Lt, which is high considering the number of possible inflection forms given a German or Lithuanian lemma. This indicates that the neural-based module does not learn generalizable inflection rules from the data effectively.

	BLEU	Lemma Usage			Term Usage		
		All	No Inf	Inf	All	No Inf	Inf
En-De Health							
AR baseline	31.9	61.2	61.1	61.6	56.7	59.6	43.0
AR w/ CD	33.4	98.6	**99.1**	96.3	82.6	**99.1**	4.5
AR w/ TLA	33.8	96.6	97.0	95.0	**89.2**	94.6	63.6
NAR baseline	31.0	56.1	56.4	54.7	52.8	55.2	41.3
NAR+C	31.1	**99.0**	**99.1**	**98.5**	82.0	**99.1**	1.4
AR w/ CD + neural	33.3	95.6	95.9	91.1	81.0	91.1	33.3
AR w/ TLA + neural	33.6	94.5	95.1	91.9	85.5	90.2	63.5
NAR+C + neural	30.9	95.6	95.8	94.9	81.1	91.1	33.8
AR w/ CD + rule	33.7	96.8	96.8	97.0	87.3	95.0	51.0
AR w/ TLA + rule	**33.9**	95.2	95.5	94.1	87.9	92.1	**68.0**
NAR+C + rule	31.7	97.1	97.0	97.5	87.1	95.0	49.5
En-Lt News							
AR baseline	14.1	64.7	76.9	42.4	55.3	74.0	21.2
AR w/ CD	13.8	89.8	**99.6**	72.0	65.2	**98.8**	3.8
AR w/ TLA	**14.4**	81.5	90.1	65.9	67.9	88.0	31.1
NAR baseline	14.3	59.4	69.0	41.7	52.7	67.8	25.0
NAR+C	14.3	89.8	99.2	72.7	64.7	98.3	3.0
AR w/ CD + neural	13.5	82.4	85.1	77.3	57.2	75.2	24.2
AR w/ TLA + neural	14.2	81.6	86.8	72.0	63.1	78.5	34.8
NAR+C + neural	14.0	83.7	88.0	75.8	58.0	77.7	22.0
AR w/ CD + rule	13.9	93.0	97.5	84.8	**75.9**	94.2	**42.4**
AR w/ TLA + rule	14.3	85.3	90.5	75.8	70.3	87.2	39.4
NAR+C + rule	14.3	**93.3**	97.1	**86.4**	75.7	94.2	41.7

Table 7.5: BLEU, lemma, and term usage rates on the En-De health and En-Lt news test suites. For lemma and term usage, we report scores on all constraints (*All*), constraints that require no inflection (*No Inf*), and constraints that require inflection (*Inf*).

End-to-End MT Evaluation Table 7.5 shows the impact of rule-based and neural-based inflection modules on top of a range of AR and NAR baselines. NAR baselines without constraints achieves competitive BLEU to the AR baseline on En-Lt and slightly lower BLEU on En-De, as in Xu and Carpuat (2021b). Given lemma constraints, AR with CD without inflection obtains lower term usage accuracy and lower BLEU than AR with TLA, as in Bergmanis and Pinnis (2021). Similar to AR with CD, NAR+C without inflection obtains lower term usage and close or lower BLEU than AR with TLA.

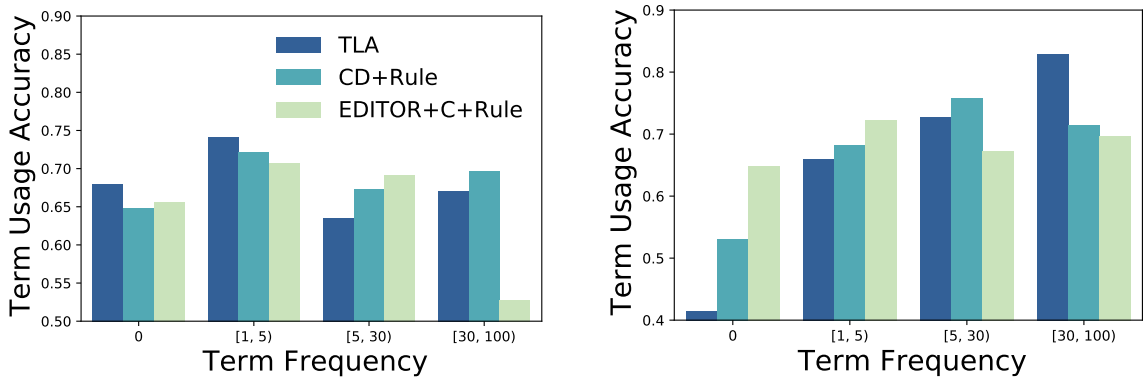
Adding rule-based inflection helps all models leverage lemma constraints more accurately.

On En-De, it significantly improves term usage accuracy of AR with CD by +4.7% and NAR+C models by +5.1%.[17] On En-Lt, it significantly improves both the lemma usage rate and term usage accuracy of AR with CD (+3.2% on lemma usage and +10.7% on term usage) and NAR+C (+3.5% on lemma usage and +11.0% on term usage). Remarkably, it also improves the term accuracy of En-Lt AR with TLA, which is already trained to inflect the target lemma constraints. When evaluating only on constraints that require inflection, the rule-based modules improves by 4.4–8.3% on TLA, 38.6–46.5% on CD, and 38.7–48.1% on NAR+C. As expected based on inflection accuracy results, rule-based modules outperform neural-based ones across the board. These improvements in term usage preserve or slightly improve BLEU.[18], as can be expected since the constraints only constitute a small portion of the tokens in the translation outputs. Overall, these results indicate that our proposed framework is model-agnostic and supports our hypothesis that the lemma constraints can be effectively inflected based on the source context alone.

We now compare our framework against TLA. Rule-based inflection combined with NAR+C achieves close lemma and term usage rates ($\Delta \leq 2\%$) to TLA on En-De, +11.8% higher lemma usage, and +7.8% higher term usage accuracy on En-Lt (the improvements are significant). On En-Lt, the largest improvements are on constraints that require inflection: +20.5% on lemma usage and +10.6% on term usage. Incorporating the constraints preserves translation quality, with no significant difference in BLEU. Overall, these results show the benefits of integrating linguistic knowledge via rule-based inflection over purely data-driven approaches. Our approach is also more adaptive, as NAR+C with rule-based inflection does not require re-training the whole

[17] All mentions of significance are based on the paired bootstrap test (Clark et al., 2011) with $p < 0.05$.

[18] The improvements on BLEU is statistically significant for NAR+C on En-De, but not for other models.



(a) En-De (b) En-Lt

Figure 7.3: Term usage accuracy of TLA, CD + rule, and NAR+C + rule binned by training set frequency.

NMT model to incorporate new lemma terms. Instead, new terms can be incorporated by updating the morphological dictionary used in the inflection module.

Cost Trade-offs Implementing the rule-based inflection module for the first target language (Lithuanian) took around 6 hours (including the time for learning the grammar knowledge from Wikipedia) by a computer scientist without prior knowledge of the target language nor formal linguistics training. The second language (German) implementation took only 3 hours, since some rules are shared across languages. By contrast, the neural-based module was implemented in about 3 hours but took around 38 hours to train a single model for one language pair on 2 GeForce GTX 1080 Ti GPUs. While these numbers do not provide a controlled comparison, they highlight that the rule-based module is relatively simple to build, as it can be done for both languages in 7-15% of the time required to train the neural model.

Term Frequency We analyze where rule-based inflection helps the most by computing the term usage accuracy on terms in different frequency bucket. As shown in Figure 7.3, the trends are different on En-De and En-Lt. On En-De, CD + rule slightly improves TLA on terms with frequency between $[5, 100)$ instead of the rare terms. One reason is that the German

source	Jim Furyk's side need eight points from Sunday's 12 singles matches to retain the **trophy**.
reference	Jimo Furyko komandai reikia gauti aštuonis taškus sekmadienio 12 vienetų mačuose, kad išsaugotų trofėjų.
constraints	**trophy**: trofėjus
reference inflection	**trophy**: trofėjų (singular, accusative)
TLA	Jim Furyk 's pusėje reikia aštuonių taškų iš sekmadienio 12 pažintys rungtynes išlaikyti trofėjus.
TLA + rule	Jim Furyk 's pusėje reikia aštuonių taškų iš sekmadienio 12 pažintys rungtynes išlaikyti trofėjų.
source	In December 2017, he was accused of assaulting his father, Todd **Palin**.
reference	2017 m. gruodžio mėnesį jis buvo apkaltintas smurtu prieš savo tėvą Toddą Paliną.
constraints	**Palin**: Palinas
reference inflection	**Palin**: Paliną (singular, accusative)
NAR+C	2017 m. gruodžio mėn. jis buvo apkaltintas užpuolimu jo tėvas Toddas Palinas.
NAR+C + rule	2017 m. gruodžio mėn. jis buvo apkaltintas užpuolęs tėvą Toddą Paliną.

Table 7.6: Translation examples comparing TLA + rule against TLA, and NAR+C + rule against NAR+C on En-Lt. We boldface the source terms with translation constraints and underline the target constraint terms used in the reference and translation outputs.

morphological dictionary that we use to determine the gender of a word and its inflection forms only covers around 70% of the constraint terms in the health test suite. In addition, NAR+C + rule underperforms CD + rule on some constraint terms with frequency between $[30, 100)$. This might be a side effect of knowledge distillation, which yields frequent errors for words that are rare in the training data (Ding et al., 2021). In En-Lt test set, 68% of the constraint terms are used in the inflection forms that are unseen in the training data. As shown in the figure, both CD + rule and NAR+C + rule bring substantial improvements over TLA on terms that are unseen in the training data. This is because most Lithuanian nouns and adjectives are inflected based on a fixed set of rules, thus even when the target lemma is unseen in the training data or morphological dictionary, it can still be inflected correctly. As a result, the rule-based inflection module can effectively incorporate linguistic knowledge in translation models and thus generalizes better to

rare and unseen terms.

Qualitative Analysis We examine a few randomly selected translation examples from TLA, NAR+C, and their counterparts with rule-based inflection. As shown in Table 7.6, TLA tends to copy constraint terms that are infrequent in the training data, and adding the rule-based inflection module helps TLA inflect the term correctly instead. In NAR+C models, the inflection module also improves the translation of the context around constraint terms, while the vanilla NAR+C model is prone to compounding errors caused by the uninflected constraints.

7.4.3 Limitations

The findings of this work should be interpreted with several limitations in mind. First, we limit our experiments to two language pairs, so it remains to be explored how the proposed framework works on languages with different degrees of inflections. Second, we evaluate models in the setting where the terminology constraints are automatically extracted and matched with the reference, and rely on automatic metrics to evaluate translation quality and the accuracy of constraint incorporation. Thus, further experiments are needed to assess how the proposed framework performs in more realistic settings where the terminology constraints are provided by end users.

7.5 Summary

We introduced a modular framework for incorporating lexical constraints in more flexible forms in NMT. The framework is based on a novel cross-lingual inflection module that inflects the target lemma constraints given source context and an NMT model that integrates the inflected

constraints in the output. We showed that our framework can be flexibly applied to different types of inflection modules, including rule-based and neural-based ones, and different NMT models, including autoregressive and non-autoregressive ones, with minimal training costs. Results on the English-German and English-Lithuanian test suites showed that the linguistically motivated rule-based inflection module helps NMT models incorporate constraint terms more accurately than both neural-based inflection and the existing end-to-end approach to incorporating lemma constraints.

Chapter 8: Conclusion and Future Work

8.1 Conclusion

This dissertation introduced training algorithms and model architectures that embed stronger inductive biases in neural sequence generation models for more sample-efficient and controllable neural machine translation (NMT). Furthermore, we investigated the inductive and spurious biases in existing models and learning algorithms for NMT, which paved the road for future studies to further improve the translation quality by strengthening the inductive biases and reducing the spurious biases.

First, we showed that by introducing stronger inductive biases through supervised learning algorithms, we can mitigate exposure bias – a well-known undesirable bias that hinders NMT models from generalizing from training to the inference scenario. We proposed a differentiable sampling algorithm to address the problem by exposing NMT models to its own predictions during training and providing reliable error signals by flexibly comparing the model's predictions with reference translations. As a result, it yields better translation quality than the standard maximum likelihood training algorithm and existing algorithms proposed to address exposure bias.

We then investigated another type of undesirable bias – the spurious biases that lead to a severe type of translation error, namely hallucinations, where the model produces a translation that is unrelated to the source. We started by identifying the internal model symptoms that flag

hallucinations triggered by source-side perturbations at inference time. Based on the discovered symptoms, we built a simple classifier which detects hallucinations on natural text inputs more accurately than strong classifiers based on quality estimation models or multilingual pre-trained language models. Finally, we connected the discovered symptoms at inference time to the spurious biases learned during training. Together, these findings suggest that we may design better training algorithms to minimize hallucinations by targeting these spurious biases.

Apart from supervised learning algorithms that mitigate undesirable biases, we also researched the inductive biases in existing semi-supervised learning algorithms – algorithms that improve the data efficiency of NMT models by integrating the language priors learned from unsupervised data into NMT without changing the model architectures. We introduced a theoretical framework that unifies existing semi-supervised learning algorithms and provided a theoretical guarantee on its global optimum. This theory explains why these algorithms work and justifies the use of iterative back-translation instead of the more complex dual learning algorithm, which is supported by the empirical study on six tasks spanning high-resource, low-resource, and cross-domain settings.

In the final part of the dissertation, we changed our focus from integrating stronger inductive biases via training algorithms to the inductive biases in model architectures toward more controllable NMT. Current NMT models are typically built upon the inductive bias that an output sequence should be predicted token by token from left to right, which makes it difficult for users to control the outputs based on their preferences and domain-specific knowledge. We addressed this problem by first introducing EDITOR, a non-autoregressive transformer model that generates a sequence through iterative insertion and reposition operations to better incorporate lexical constraints in NMT outputs. We also presented the novel dual-path imitation learning algorithm to

train the model. Experiments on three translation tasks showed that EDITOR incorporates lexical constraints more effectively than the state-of-the-art edit-based NMT model, and more efficiently than the traditional autoregressive model. To allow users to specify lexical constraints in their lemma forms without inflections, we further extended this work with a modular framework that consists of a novel cross-lingual inflection module and an NMT model. Based on this framework, different types of inflection modules and NMT models can be flexibly combined together with minimal computational overhead. Evaluation results on two test suites showed that this framework combined with a linguistically motivated rule-based inflection module incorporates lemma-form constraints more accurately than the existing end-to-end approach.

8.2 Future Work

We wrap up the dissertation by discussing its limitations and directions for future research.

8.2.1 Mitigating Hallucinations in Machine Translation and Other Tasks

In Chapter 4, we revealed the spurious biases that are related to the hallucination phenomenon in MT – for some training samples, the model learns to predict the ground-truth sequence by memorizing it while ignoring a large part of the source sentence. This suggests approaches to mitigating hallucinations in MT by targeting these spurious biases. For example, we could monitor the source contributions to the model's prediction on ground-truth tokens using interpretation techniques (such as saliency maps (Bach et al., 2015; Simonyan et al., 2013) and input perturbations (Feng et al., 2018; Li et al., 2016b)) at training time and down-weight the gradient updates on samples where the model exhibits the aforementioned

spurious biases. We could also design adversarial learning algorithms (Belinkov et al., 2019; Ramakrishnan et al., 2018) to combat the spurious biases. Furthermore, we would like to examine whether the hallucination symptoms discovered on MT generalize to other language generation tasks, including abstractive summarization (Falke et al., 2019; Maynez et al., 2020b), dialogue generation (Dušek et al., 2018), and data-to-text generation (Wiseman et al., 2017). This could open the door to universal approaches to detecting and mitigating hallucinations on a wide range of language generation tasks.

8.2.2 Inductive Biases in Large Pre-trained Language Models

We studied the inductive biases introduced through the semi-supervised learning algorithms that train NMT models jointly on supervised and unsupervised data in Chapter 5. Recent advances in large pre-trained language models (Brown et al., 2020; Devlin et al., 2019; Liu et al., 2020b; Zhang et al., 2022) have revealed another way to leverage the unsupervised data – by pre-training large language models on unsupervised text corpora using self-supervised learning objectives and adapting it to downstream tasks through supervised fine-tuning or prompting. This training paradigm has brought improved performance on a wide range of language tasks including machine translation, question answering, reading comprehension, and natural language inference (Brown et al., 2020; Devlin et al., 2019; Roberts et al., 2020). Additionally, such pre-trained models have been shown to generalize better to the challenging datasets designed to counter specific spurious biases (Tu et al., 2020) and achieve strong performance in few-shot settings (Brown et al., 2020). However, generating texts that are truthful and faithful to the source remains a challenge for large pre-trained language models, and scaling up models alone does not

always solve the problem – sometimes it worsens it (Lin et al., 2022). To address this issue, we need to first understand what inductive biases are introduced through pre-training. We would like to investigate what is learned during the pre-training phase, and what is being transferred from pre-training to the target tasks. Especially in the context of machine translation, future work is needed to explain why models pre-trained on English monolingual corpora can achieve reasonable performance on machine translation given a few translation examples.

8.2.3 Edit-based Generation for a Broader Range of Tasks

While the main focus of this dissertation is on machine translation, the EDITOR model introduced in Chapter 6 is a generic model that can be adapted to a wide range of language generation tasks that may benefit from controlled generation through explicit editing. For example, EDITOR has shown outstanding performance on text simplification (Agrawal et al., 2021) and abstractive summarization (Agrawal and Carpuat, 2022). We would like to further apply EDITOR to other text editing tasks such as post-editing and style transfer.

Additionally, many text generation problems require generating long, coherent text (e.g. document-level machine translation, story generation, multi-document summarization). Prior work showed that autoregressive models still suffer from syntactic and semantic errors when generating long passages of text (Tan et al., 2021). On the other hand, EDITOR has the potential to generate more coherent, error-free text, as it allows for flexible revising and editing on the generated text. Thus, it would be an interesting future direction to adapt EDITOR to these tasks. One remaining challenge would be how to train the model in a sample-efficient way since many of the long text generation problems suffer from the data scarcity issue.

Bibliography

Mostafa Abdou, Vladan Glončák, and Ondřej Bojar. 2017. Variable mini-batch sizing and pre-trained embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 680–686, Copenhagen, Denmark. Association for Computational Linguistics.

Fadi Abu Sheikha and Diana Inkpen. 2011. Generation of formal and informal sentences. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 187–193, Nancy, France. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. 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 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2022. An imitation learning curriculum for text editing with non-autoregressive models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7550–7563, Dublin, Ireland. Association for Computational Linguistics.

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769, Online. Association for Computational Linguistics.

Duygu Altinok. 2018. Demorphy, german language morphological analyzer. *CoRR*, abs/1803.00902.

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations*.

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *CoRR*, abs/1607.07086.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3th International Conference on Learning Representations*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. 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 673–681, Minneapolis, Minnesota. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28*, pages 1171–1179.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. 2019a. Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019b. Naver labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer, Harry Printz, and Lubos Ures. 1994. The Candide system for machine translation. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Yotam Berger. 2017. Israel arrests Palestinian because Facebook translated 'good morning' to 'attack them' - Israel News - Haaretz.com. https://www.haaretz.com/israel-news/palestinian-arrested-over-mistranslated-good-morning-facebook-post-1.5459427.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondrej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation*, pages 272–307. Association for Computational Linguistics.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Niccolò Campolungo, Tommaso Pasini, Denis Emelin, and Roberto Navigli. 2022. Reducing disambiguation biases in NMT by leveraging explicit word sense information. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4824–4838, Seattle, United States. Association for Computational Linguistics.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alipio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017. Overview of the IWSLT 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The IWSLT 2015 evaluation campaign. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–14, Da Nang, Vietnam.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. 2015. Learning to search better than your teacher. In *International Conference on Machine Learning*, pages 2058–2066. PMLR.

Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. The University of Edinburgh's English-German and English-Hausa submissions to the WMT21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 104–109, Online. Association for Computational Linguistics.

Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. 2018. Fast policy learning through imitation and reinforcement. In *Proceedings of the 2018 Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 845–855, Monterey, CA, USA.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5094–5101. AAAI Press.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, Montreal, Canada.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

William W. Cohen and Victor R. Carvalho. 2005. Stacked Sequential Learning. In *Proceedings of the IJCAI*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell and Julia Kreutzer. 2018. Explaining and generalizing back-translation through wake-sleep. *CoRR*, abs/1806.04402.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(Jan):951–991.

Anna Currey and Kenneth Heafield. 2019. Incorporating source syntax into transformer-based neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 24–33, Florence, Italy. Association for Computational Linguistics.

Hal Daumé, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3):297–325.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine learning*, 75(3):297–325.

Thomas Degris, Patrick M Pilarski, and Richard S Sutton. 2012. Model-free reinforcement learning with continuous action in practice. In *American Control Conference (ACC), 2012*, pages 2177–2182. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. 2019. Avoiding latent variable collapse with generative skip models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2397–2405.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Understanding and improving lexical choice in non-autoregressive translation. In *International Conference on Learning Representations*.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven Word Alignment Interpretation for Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 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.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Daniel Edmiston. 2020. A systematic analysis of morphological content in BERT models for multiple languages. *CoRR*, abs/2004.03032.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada. Association for Computational Linguistics.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 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.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Katja Filippova. 2020. Controlled Hallucinations: Learning to Generate Faithfully from Noisy Data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870, Online. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised Quality Estimation for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

George Foster, Philippe Langlais, and Guy Lapalme. 2002. User-friendly text prediction for translators. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 148–155. Association for Computational Linguistics.

Mercedes García-Martínez, Ozan Caglayan, Walid Aransa, Adrien Bardet, Fethi Bougares, and Loïc Barrault. 2017. LIUM machine translation systems for WMT17 news translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 288–295. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR.

Ulrich Germann. 2020. The University of Edinburgh's submission to the German-to-English and English-to-German tracks in the WMT 2020 news translation and zero-shot translation robustness tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 197–201, Online. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2017. Differentiable scheduled sampling for credit assignment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 366–371. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc.

Nuno M. Guerreiro, Elena Voita, and André F. T. Martins. 2022. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 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.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic Chinese to English news translation. *CoRR*, abs/1803.05567.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems 29*, pages 820–828. Curran Associates, Inc.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Shilin He, Zhaopeng Tu, Xing Wang, Longyue Wang, Michael Lyu, and Shuming Shi. 2019. Towards understanding neural machine translation with word importance. 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 953–962, Hong Kong, China. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages.

Jindřich Helcl, Barry Haddow, and Alexandra Birch. 2022. Non-autoregressive machine translation: It's not as fast as it seems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1780–1790, Seattle, United States. Association for Computational Linguistics.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *CoRR*, abs/1712.05690.

M. Hledík, T. R. Sokolowski, and G. Tkačik. 2019. A tight upper bound on mutual information. In *2019 IEEE Information Theory Workshop (ITW)*, pages 1–5.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Matthew D Hoffman and Matthew J Johnson. 2016. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *Proceedings of NIPS Workshop in Advances in Approximate Bayesian Inference*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*, volume 398. John Wiley & Sons.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Matti Kääriäinen and John Langford. 2006. Lower Bounds for Reductions. In *Talk at the Atomic Learning Workshop*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3th International Conference on Learning Representations*, San Diego, CA, USA.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled Multi-Task Learning: From Syntax to Translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. *CoRR*, abs/2107.10821.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Conference of the Association for Machine Translation in the Americas*, pages 115–124. Springer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 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.

Taku Kudo and John Richardson. 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, Brussels, Belgium. Association for Computational Linguistics.

Laura Kurtzman. 2019. Google translates doctor's orders into Spanish and Chinese with few significant errors - UCSF News. https://www.ucsf.edu/news/2019/02/413376/google-translates-doctors-orders-spanish-and-chinese-few-significant-errors.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the 6th International Conference on Learning Representations*.

John Langford and Bianca Zadrozny. 2005. Relating reinforcement learning performance to classification performance. In *Proceedings of the 22nd international conference on Machine learning*, pages 473–480. ACM.

Rémi Leblond, Jean-Baptiste Alayrac, Anton Osokin, and Simon Lacoste-Julien. 2018. SEARNN: Training RNNs with global-local losses. In *International Conference on Learning Representations*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018a. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018b. Hallucinations in neural machine translation. In *NeurIPS Interpretability and Robustness for Audio, Speech and Language Workshop*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019a. Towards explainable NLP: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581, Florence, Italy. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019b. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. 2019. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. 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 4282–4292, Hong Kong, China. Association for Computational Linguistics.

Xutai Ma, Ke Li, and Philipp Koehn. 2018. An analysis of source context dependency in neural machine translation. In *21st Annual Conference of the European Association for Machine Translation*, page 189.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In *Advances in Neural Information Processing Systems 27*, pages 3086–3094. Curran Associates, Inc.

Marianna Martindale, Marine Carpuat, Kevin Duh, and Paul McNamee. 2019. Identifying Fluently Inadequate Output in Neural and Statistical Machine Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 233–243, Dublin, Ireland. European Association for Machine Translation.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020a. 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.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020b. 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.

Aingeru Mayor, Inaki Alegria, Arantza Díaz de Ilarraza, Gorka Labaka, Mikel Lersundi, and Kepa Sarasola. 2011. Matxin, an open-source rule-based machine translation system for basque. *Machine translation*, 25(1):53–82.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 720–730, San Diego, California. Association for Computational Linguistics.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328, Austin, Texas. Association for Computational Linguistics.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010a. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, et al. 2010b. Recurrent neural network based language model. In *In INTERSPEECH 2010,*. Citeseer.

Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ....

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2019. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. NTT's neural machine translation systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 461–466, Belgium, Brussels. Association for Computational Linguistics.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Hideto Kazawa, Yusuke Oda, Graham Neubig, and Sadao Kurohashi. 2017. Overview of the 4th workshop on Asian translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 1–54, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

Khanh Nguyen and Hal Daumé III. 2019. Global Voices: Crossing borders in automatic news summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 90–97, Hong Kong, China. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2018. Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 334–343. Association for Computational Linguistics.

Jan Niehues and Eunah Cho. 2017. Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proceedings of the Second Conference on Machine Translation*, pages 80–89, Copenhagen, Denmark. Association for Computational Linguistics.

Xing Niu, Michael Denkowski, and Marine Carpuat. 2018. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 84–91. Association for Computational Linguistics.

Xing Niu, Weijia Xu, and Marine Carpuat. 2019. Bi-directional differentiable input reconstruction for low-resource neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Douglas W Oard. 1998. A comparative study of query and document translation for cross-language information retrieval. In *Conference of the Association for Machine Translation in the Americas*, pages 472–483. Springer.

Douglas W Oard, Marine Carpuat, Petra Galuščáková, Joseph Barrow, Suraj Nair, Xing Niu, Han-Chin Shing, Weijia Xu, Elena Zotkina, Kathleen McKeown, et al. 2019. Surprise languages: rapid-response cross-language ir. In *Proceedings of the Ninth International Workshop on Evaluating Information Access (EVIA 2019)*, page 23.

Franz Josef Och. 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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jessica Ouyang, Boya Song, and Kathy McKeown. 2019. A robust abstractive system for cross-lingual 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 2025–2031, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 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.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. 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 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Computational*, pages 157–163. Association for Computational Linguistics.

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

Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming language priors in visual question answering with adversarial regularization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The Curious Case of Hallucinations in Neural Machine Translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Matīss Rikters and Mark Fishel. 2017. Confidence Through Attention. In *Proceedings of the 16th Machine Translation Summit (MT Summit 2017)*, Nagoya, Japan.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stéphane Ross and J. Andrew Bagnell. 2014. Reinforcement and imitation learning via interactive no-regret learning. *CoRR*, abs/1406.5979.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635. PMLR.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Sashank Santhanam, Behnam Hedayatnia, Spandana Gella, Aishwarya Padmakumar, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tur. 2021. Rome was built in 1776: A case study on factual correctness in knowledge-grounded response generation. *CoRR*, abs/2110.05456.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 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.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Ivan Skorokhodov, Anton Rykachevskiy, Dmitry Emelyanenko, Sergey Slotin, and Anton Ponkratov. 2018. Semi-supervised neural machine translation with language models. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, pages 37–44, Boston, MA. Association for Machine Translation in the Americas.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 204–211, Brussels, Belgium. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985, Long Beach, California, USA. PMLR.

Katsuhito Sudoh, Kosuke Takahashi, and Satoshi Nakamura. 2021. Is this translation error critical?: Classification-based human and automatic machine translation evaluation focusing on critical errors. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 46–55, Online. Association for Computational Linguistics.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Kosuke Takahashi, Yoichi Ishibashi, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Multilingual machine translation evaluation metrics fine-tuned on pseudo-negative examples for WMT 2021 metrics task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1049–1052, Online. Association for Computational Linguistics.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324, Online. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. 2018. Parallel WaveNet: Fast high-fidelity speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926, Stockholmsmässan, Stockholm Sweden. PMLR.

Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, volume 2, page 5. AAAI Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, Long Beach, CA, USA. Curran Associates, Inc.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3024–3030. AAAI Press.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*, Lille, France.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. 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 1126–1140, Online. Association for Computational Linguistics.

Chaojun Wang and Rico Sennrich. 2020. On Exposure Bias, Hallucination and Domain Shift in Neural Machine Translation. *arXiv:2005.03642 [cs]*.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Non-autoregressive machine translation with auxiliary regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5377–5384.

Sean Welleck, Kianté Brantley, Hal Daumé III, and Kyunghyun Cho. 2019. Non-monotonic sequential text generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6716–6726. PMLR.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yijun Xiao and William Yang Wang. 2021a. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021b. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Weijia Xu and Marine Carpuat. 2018. The University of Maryland's Chinese-English neural machine translation systems at WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 535–540, Belgium, Brussels. Association for Computational Linguistics.

Weijia Xu and Marine Carpuat. 2021a. EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

Weijia Xu and Marine Carpuat. 2021b. EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

Weijia Xu and Marine Carpuat. 2021c. Rule-based morphological inflection improves neural terminology translation.

Weijia Xu, Xing Niu, and Marine Carpuat. 2019. Differentiable sampling with flexible reference word order for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2047–2053, Minneapolis, Minnesota. Association for Computational Linguistics.

Weijia Xu, Xing Niu, and Marine Carpuat. 2020. Dual reconstruction: a unifying objective for semi-supervised neural machine translation. In *Findings of the Association for Computational*

Linguistics: EMNLP 2020, pages 2006–2020, Online. Association for Computational Linguistics.

Elizaveta Yankovskaya, Andre Tättar, and Mark Fishel. 2018. Quality estimation with force-decoded attention and cross-lingual embeddings. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 816–821, Belgium, Brussels. Association for Computational Linguistics.

Poorya Zaremoodi and Gholamreza Haffari. 2018. Neural machine translation for bilingually scarce scenarios: a deep multi-task learning approach. 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 1356–1365, New Orleans, Louisiana. Association for Computational Linguistics.

Meishan Zhang, Zhenghua Li, Guohong Fu, and Min Zhang. 2019. Syntax-enhanced neural machine translation with syntax-aware word representations. 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 1151–1161, Minneapolis, Minnesota. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 555–562. AAAI Press.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

Chunting Zhou, Xuezhe Ma, Junjie Hu, and Graham Neubig. 2019. Handling syntactic divergence in low-resource machine translation. 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 1388–1394, Hong Kong, China. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.