Gated Transformer for Robust De-noised Sequence-to-Sequence Modelling

Ayan Sengupta[1], **Amit Kumar**[1], **Sourabh Kumar Bhattacharjee**[2]*, **Suman Roy**[1]
[1]Optum Global Advantage (UnitedHealth Group), India
[2]New York University, USA
{ayan_sengupta, amit.kumar, suman.roy}@optum.com;
skb5275@nyu.edu

Abstract

Robust sequence-to-sequence modelling is an essential task in the real world where inputs are often noisy. Both user-generated and machine generated inputs contain various kinds of noises in the form of spelling mistakes, grammatical errors, character recognition errors etc, all of which impact downstream tasks and affect interpretability of texts. In this work, we devise a novel sequence-to-sequence architecture for detecting and correcting different real world and artificial noises (adversarial attacks) from English texts. Towards that we propose a modified transformer-based encoder-decoder architecture that uses a gating mechanism to detect types of corrections required and accordingly corrects texts. Experimental results show that our gated architecture with pre-trained language models perform significantly better that the non-gated counterparts and other state-of-the-art error correction models in correcting spelling and grammatical errors. Extrinsic evaluation of our model on Machine Translation (MT) and Summarization tasks show the competitive performance of the model against other generative sequence-to-sequence models under noisy inputs.

1 Introduction

Noisy texts are very common in user-generated texts that appear abundant in various social media platforms like short message service (SMS), online chat, email, blogs, wikis etc. These kind of texts may contain spelling errors, abbreviations, non-standard terminology, false starts to name a few. Most of the NLP models assume the data to be linguistically correct and semantically coherent. Thus, noisy texts pose a serious threat in ensuring accurate predictions and practicality of any NLP system in real-life applications. Automatic noise correction from texts is thus crucial in many systems such as user provided search (Gao et al., 2010), social media analysis (Baldwin et al., 2013; Mapa et al., 2012), customer feedback analysis etc. As described by Keselj *et al.* (Keselj, 2009), each human-typed text contain 1-2% spelling and grammatical errors and 10-15% of them are from web searches. Other sources of noises can originate from machine extracted outputs such as optical character recognition (OCR) (Pontes et al., 2019; Mutuvi et al., 2018; Wang et al., 2018) and speech-to-text generation (Guo et al., 2019; Bassil and Alwani, 2012) which need to be corrected in order to improve the performance on downstream tasks.

To devise a robust technique for noise removal and corrected target generation, we propose `gated-Trans` (g-Trans), a gated transformer sequence-to-sequence model. Our model uses a *mask gate* based on a pre-trained transformer encoder to detect noises within texts, and a pre-trained transformer decoder to generate noise-free target sequence. The decoder uses a *copy gate* to determine whether to copy an output token directly from the input, a *generate gate* to generate new output token for a masked (contextually incorrect) token and a *skip gate* to skip tokens that are contextually irrelevant. We evaluate our model on real-life noisy texts generated from OCR engines, as well as artificial augmentation based noises (Ma, 2019; Morris et al., 2020) that replicate the real-life user generated noises. Further, our extrinsic evaluation on the noisy machine translation (MT) and summarization tasks shows the robustness of our model on translating noisy texts into correct generated target.

Contribution: We contribute to the existing body of noise removal task in several ways. • We introduce a gating mechanism for conditional target generation from transformer; • The method introduced is robust to different noising techniques which makes it adaptable for real-life noise correction; • Our proposed method can also handle noises injected during pre- and post-tokenization

[*] The author was employed at Optum Global Advantage, India during the entire work.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3645–3657
November 7–11, 2021. ©2021 Association for Computational Linguistics

phase; • The proposed model can efficiently handle noisy texts in extrinsic tasks like machine translation, text summarization etc, occurring in real-life applications.

Reproducibility: Source codes and other experimental details to reproduce the results have been made public at `https://github.com/victor7246/gated-Transformer`. The datasets are enclosed in the supplementary material.

2 Related Work

In this work we adopt the definition of noise in text from (Contractor et al., 2010) as any kind of difference between the surface form of a coded representation of the text and the correct text. As more and more noisy text data being generated in various social communication media, removing noises from these texts have become an increasingly important task. Existing methodologies for noise removal from texts can be divided into two categories - classifier based approaches and statistical machine translation (SMT) based approaches. Traditional classifier-based approaches (Imamura et al., 2003; Khadivi and Ney, 2005) and fine-tuned SMT based methods (Junczys-Dowmunt and Grundkiewicz, 2016; Hoang et al., 2016; Chollampatt et al., 2016) are not generalized enough to correct different types of noises from texts (Ng et al., 2014) and require huge parallel corpora. Thus in SMT-based approaches usage of language models pre-trained on monolingual corpora has become very popular. Etoori *et al.* (Etoori et al., 2018) propose a Seq2Seq-based deep learning model to perform spell correction automatically in resource-scare languages like Hindi and Telegu. Krishna *et al.* (Krishna et al., 2018b) propose a post-OCR text correction approach based on Seq2Seq model for digitising texts using Romanised Sanskrit, the lack of resources has made them use OCR models trained for other languages written in Roman. Wang *et al.* use Confusion-set-guided Pointer Networks (Wang et al., 2019), a novel Seq2Seq model for the task of Chinese Spell Correction (CSC).

Researchers have recently tried using large pre-trained transformer language models to capture semantic understanding of texts accurately and achieve extremely competitive performance across various NLP tasks including spell correction. Hong *et al.* (Hong et al., 2019) have recently proposed BERT (Devlin et al., 2019) based Seq2Seq model for the task of CSC. However as pointed out by (Zhang et al., 2020), vanilla BERT is difficult to use for spelling correction, as it is primarily a pre-trained masked language model (MLM). The authors in (Lewis et al., 2020) have proposed a large pre-trained transformer model, BART, which is a denoising sequence-to-sequence language model. (Malmi et al., 2019) propose a BERT based encoder-decoder model to correcting texts with edit operations. Similar transformer models have been successfully used in Grammatical Error Correction (GEC) systems. For example, authors in (Omelianchuk et al., 2020) employs a transformer encoder to design a simple and efficient GEC sequence tagger called GECToR. In (Kaneko et al., 2020) a pre-trained masked language model like BERT is effectively incorporated into an encoder-decoder model in a GEC system. People have also investigated the quality of output produced by GEC systems, - researchers proposes a neural approach (Chollampatt and Ng, 2018) to automatically estimate the quality of GEC-produced sentences which do not use any hand-crafted features.

Noises can be introduced into texts in two phases, pre-tokenization and post-tokenization. BART uses *masking*, *deletion* and *infilling* for introducing noises in pre-training. Similarly, T5 (Raffel et al., 2020) is another pre-trained language model that uses fill-in-the-blank-style denoising objective for pre-training. However, in both BART and T5 the noises are added only after the tokenization of texts. So, denoising raw noisy texts is assumed to be more difficult than denoising noisy token sequence. Further, none of the existing pre-trained language models are evaluated on noisy generation tasks with corrupted source texts.

To overcome these limitations, in this work we introduce a robust transformer-based model that can detect and remove noises from texts injected during pre-tokenization and generate correct target texts flexibly. Further, using different gating strategies, our model can understand the different kinds of induced and natural noises and act differently under various scenarios, which is not offered by BART, T5 or other pre-trained denoising language models.

3 The Proposed Model

In this section we formally describe our model `gated-Trans` that consists of a transformer based gated-encoder and a gated-decoder. To showcase the efficacy of our proposed gating mechanism over the existing pre-trained language models, in this work we use the pre-trained BERT, BART and T5 models as our backbone to initialize the encoder and decoder layers. In the subsequent discussion we assume each noisy source text is tokenized into m subword tokens represented as a sequence of vectors $\mathbf{X}_{1:m} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, where each $\mathbf{x}_i, 1 \leq i \leq m$ represents the vector representation of the ith token. Given a sequence of tokens $\mathbf{X}_{1:m}$ the goal is to transform it into another sequence of subword tokens $\mathbf{Y}_{1:n} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$, whereas the length of target vectors n is unknown apriori and depends on the input sequence. The denoising process in BART and T5 is done after tokenization. So, an incorrect token (subword) is replaced with another token. On the other hand, in our gated-transformer, we use the raw, noisy text, tokenize them and feed them onto the sequence-to-sequence model. This task is assumed to be more challenging, as an incorrect word can be tokenized into multiple incorrect tokens (subword) after tokenization. Hence, there may not be a one-to-one correspondence between the input and the output. Also, as it is difficult to establish an one-to-one correspondence between source and target text, contrary to Chinese spell correction task, we adhere to word-level and subword-level text denoising. In this work we adapt byte pair encoding (BPE) (Sennrich et al., 2016) to convert both noisy input and the correct target into sequences of subwords. In the following subsections we describe each of the constituent modules of our model in greater detail (also see Figure 1 for the architecture of the proposed model).

3.1 Masked Encoder

As described in previous section, we use pre-trained language models to initialize the weights of our encoder. The encoder layer consists of a fixed number L identical transformer blocks each of which uses a fixed K number of self-attention heads and d-dimensional feed-forward dense layers. For example, `gated-Trans` with BERT backbone model consists of 12 encoder layers (similar to BERT-base architecture) with each having 12 self-attention heads and 768-dimensional FFN layers. Hence, each layer l generates a hidden representation $\mathbf{h}^{(l)} = (\mathbf{h}_1^{(l)}, \mathbf{h}_2^{(l)}, \cdots \mathbf{h}_m^{(l)})$, which is generated using the Multi-headed Self-attention (MHA) and FFN at layer l. For each token $\mathbf{x}_i$, we use the hidden state $\mathbf{h}_i^{(L)}$ obtained from L^{th} layer to calculate the masked hidden state using:

$$\mathbf{u}_i = \sigma(\mathbf{W}_{mask} \cdot \mathbf{h}_i^{(L)}) \tag{1}$$

and construct $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_m)$. Further

$$\mathbf{h} = \mathbf{emb}_{mask} \cdot \mathbf{U} + (\mathbf{1}_m - \mathbf{U}) \odot \mathbf{h}^{(L)} \tag{2}$$

Above σ is the sigmoid activation function and $\mathbf{1}_m$ denotes an m-dimensional vector of 1's. Also, $\mathbf{emb}_{mask}$ is the embedding of the $[MASK]$ token from the corresponding encoder model and $\odot$ is the element-wise multiplication. The *masking probability* $\mathbf{u}_i$ determines whether we need to explicitly replace the token $\mathbf{x}_i$ with the $[MASK]$ token. This is very similar to the soft-masking proposed in (Zhang et al., 2020). However, soft-masking probability is calculated by a separate detection network under the supervision of labels corresponding to detection task.

3.2 Conditional Decoder

In the decoder, we intend to calculate the probability $p_{\theta_{dec}}(\mathbf{Y}_{1:n}|\mathbf{X}_{1:m})$. By Bayes' rule we can decompose this probability in an auto-regressive manner into conditional probabilities of single target vectors being conditioned on the decoder inputs.

$$P_{\theta_{dec}}(\mathbf{Y}_{1:n}|\mathbf{X}_{1:m}) = \prod_{i=1}^{n} P_{\theta_{dec}}(\mathbf{y}_i|\mathbf{Y}_{0:i-1}, \mathbf{X}_{1:m})$$

We initialize $\mathbf{y}_0$ with the $[CLS]$ or, $[START]$ token. Similar to the encoder, the decoder calculates self-attention among the decoder hidden states. However, in decoding phase we have *uni-directional* self-attention among decoder tokens and *cross-attention* between decoder states and encoder hidden states. Similar to the self-attention operation in encoder, for decoder we project the embeddings of a token $\mathbf{y}_i^{'}$ ($\mathbf{y}_{i-1}$ shifted right) to query, key and value triplets. In uni-directional self-attention we calculate dot product attention using q_i as queries, $(k_0, k_1, \cdots k_i)$ as keys and $(v_0, v_1, \cdots, v_i)$ as values, all of which are projected from $\mathbf{y}_i^{'}$. However, in cross-attention, we use $\mathbf{y}_i^{'}$ to project to query and $\mathbf{X}_{1:m}$ to project to keys and values. For each token $\mathbf{y}_i^{'}$, we consider the hidden states $\mathbf{h}_i^{'}$ from the last layer of decoder

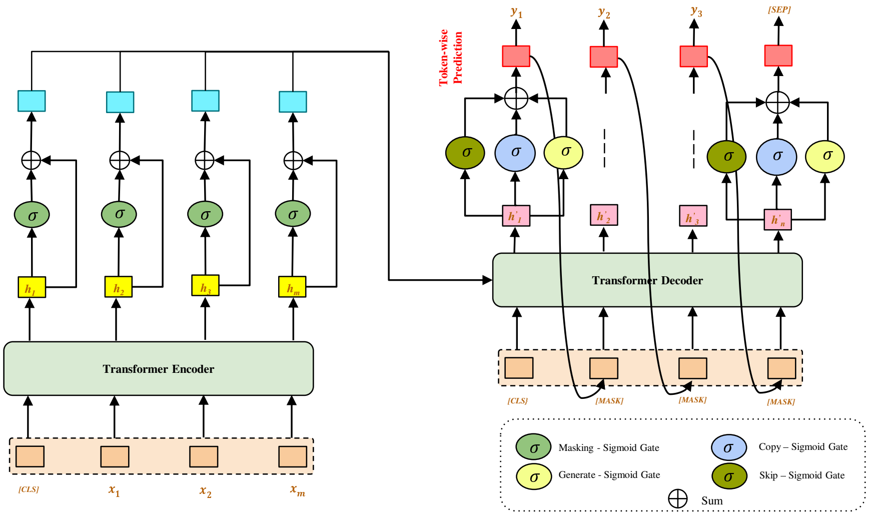


Figure 1: `gated-Trans` using gated pre-trained transformer encoder-decoder model

block and calculate the *copy probability*, *generate probability* and *skip probability* respectively as follows:

$$\mathbf{c}_i = \sigma(\mathbf{W}_{copy} \cdot \mathbf{h}_i^{'}) \tag{3}$$

$$\mathbf{g}_i = \sigma(\mathbf{W}_{gen} \cdot \mathbf{h}_i^{'}) \tag{4}$$

$$\mathbf{s}_i = \sigma(\mathbf{W}_{skip} \cdot \mathbf{h}_i^{'}) \tag{5}$$

Subsequently, we normalize the gate probabilities:

$$\mathbf{c}_i, \mathbf{g}_i, \mathbf{s}_i := softmax([\mathbf{c}_i, \mathbf{g}_i, \mathbf{s}_i]) \tag{6}$$

Intuitively, for a masked token in the encoder, we should have a high probability assigned to the generate gate in order to generate a new token instead of the incorrect token. On the other hand, correct tokens (where mask probability is low) need to be copied directly. The intuition behind using this copy gate is similar to the concept of Copy-Net (Gulcehre et al., 2016; Wang et al., 2019; See et al., 2017). The *skip gate* is introduced to tackle insertion based text attacks. For contextually incorrect word, if the model expects low probability for the generation of a new token, and low probability for copying the incorrect token, it can skip the token altogether. Finally we update the hidden state with

$$\mathbf{h}_i^{'} = \mathbf{c}_i \cdot \mathbf{y}_i^{'} + \mathbf{g}_i \cdot \sigma(\mathbf{h}_i^{'}) + \mathbf{s}_i \cdot \mathbf{emb}_{mask} \tag{7}$$

In the last layer of our conditional decoder we use a softmax activation function to project the hidden states to obtain most probable candidate for the generation. We use

$$\begin{aligned} &P_{\theta_{dec}}(\mathbf{y}_i | \mathbf{Y}_{0:i-1}, \mathbf{X}_{1:m}) \\ &\qquad = softmax(\mathbf{W}_{vocab} \cdot \mathbf{h}_i^{'} + \mathbf{b}_{vocab}) \end{aligned} \tag{8}$$

3.3 Learning

Unlike in (Zhang et al., 2020), we train `gated-Trans` end-to-end based on the generation task. Hence, for a given noisy text $text_i^{noisy}$ and the target text $text_i^{target}$, we tokenize the input noisy text to generate a noisy input sequence $\mathbf{X}_{1:m} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ and the target ground truth sequence $\mathbf{Y}_{1:n} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ and calculate the loss as:

$$\mathcal{L}_i = \sum_{j=1}^{n} \mathbf{y}_j \log(P_{\theta_{dec}}(\mathbf{y}_j | \mathbf{Y}_{0:j-1}, \mathbf{X}_{1:m})) \tag{9}$$

During training we calculate the loss $\mathcal{L}_i$ for the entire mini-batch to learn the parameters for both encoder as well as, the decoder.

4 Experiments and Results

In this section we describe our experimental efforts on both intrinsic and extrinsic evaluations and subsequently report the results.

4.1 Dataset

In this study we use total 5 datasets for evaluating our model against the baselines. We divide our experimental study into two parts, - *intrinsic evaluation* and *extrinsic evaluation*. We report the statistics of these datasets in Table 1. Additionally, we also report the average error percentage in each text for each of these corpora in the form of **Word Recognition Rate** (WRR) (Krishna et al., 2018a). WRR denotes the percentage of correct words present in the noisy input text.

	Dataset	#Sent	Source		Target		Noise %
			Length	#Token	Length	#Token	
OCR	ALTA	6,000	471	372,025	476	326,626	19.83
	ICDAR	765	326	41,411	314	38,629	13.11
Infused	IMDb	50,000	329	449,582	231	438,729	29.17
	WMT14	10,000	22	42,619	21	37,398	40.72
	CNN/DM	10,000	220	120,101	35	46,354	39.50

Table 1: Dataset statistics. Noise is defined as $1 - WRR$ (Krishna et al., 2018a).

Intrinsic Evaluation Datasets: In this work, we showcase the robustness of our model on machine-generated noises (e.g. OCR) and real-life noises such as - random insertion, deletion, swapping. We use 3 different datasets, containing original and noisy text pairs, for intrinsic evaluation described as below.

• **ALTA:** We use the dataset collected by (Molla and Cassidy, 2017) for ALTA 2017 shared task 1 which consists of original output of OCR system for each of the documents, along with their corrected versions.

• **ICDAR:** We use the post-OCR correction dataset introduced in ICDAR 2017 and 2019 competitions (Chiron et al., 2017; Rigaud et al., 2019), which have been curated mostly from English newspaper and monographs.

• **IMDb:** Additionally we use a corpus of movie reviews from IMDb which is collected by (Maas et al., 2011) and primarily used for sentiment classification. The primary reason to use the IMDb dataset is to use a standard real-life text dataset for injecting real-life artificial noises and reconstruct the original text to showcase the robustness of our model for real-life applications. We use character level and contextual text augmentation techniques (described in A.2) externally using nlpaug[1] for noise injections.

Extrinsic Evaluation Datasets: We perform our extrinsic evaluation on two tasks - (1) Machine Translation (MT) and (2) Summarization, and inject artificial noises in the source text of the following datasets.

• **WMT14:** We utilize a subset of WMT14 English-French dataset (Bojar et al., 2014) for machine translation task and inject artificial noises to the English source text to understand the capability of our model to translate a noisy English text to the coherently translated French text.

• **CNN/DM:** We use a subset of the dataset used in CNN/DailyMail news summarization (Hermann et al., 2015). Similar to MT, noises were injected to the source text and evaluate against the predicted summary.

Augmentation Type	**Augmented Text**
Random	**T3he** quick brown **fEox** jumps over **th6e la1y d*og**
Keyboard	The quick brown **Gox juJps ocer** the lazy dog
Swap	**Hte quikc borwn** fox jumps **ovre teh** lazy **dgo**
Delete	**Te quic** brown **fx** jumps **ver he laz og**

Table 2: Examples of Injected Noise Augmentations. The original text is *“The quick brown fox jumps over the lazy fox"*. Out of all these techniques, only *Random* augmenter does insertion based noise injection.

4.1.1 Noises in Datasets

• **Noises in Intrinsic Evaluation Datasets:** The intrinsic experimental datasets contain OCR extracted errors which comprise of variations of spelling errors. These errors can be categorized into - *multi-token errors*, *first-position errors* and *run-on errors*. We describe each of these types of errors in greater details in the appendix.

• **Infused Noises:** We explore different adversarial attack based noises to infuse noises artificially to the source texts of IMDb, MT and summarization datasets. We use *Random Character*, *Keyboard*, *Character Swap* and *Character Deletion* based augmentation techniques. *Random noise*, is a combination of all three noises where we observe character insertion, deletion, and swapping together within a single sentence. A list of examples of augmented noises are shown in Table 2. We provide the further details on these augmentation techniques in the appendix.

4.2 Baseline Methods

In this work we adopt several state-of-the-art models for spelling, grammatical and noise correction.

▷ **Symspell:** As explored by (Stahlberg et al., 2019), we use Symspell[2], a simple spell checker based on confusion sets as the simplest baseline. It uses a handcrafted confusion set (Bryant and Briscoe, 2018) as a lookup dictionary to correct incorrect words.

▷ **GECToR** (Omelianchuk et al., 2020) : This model uses a transformer encoder based efficient GEC sequence tagging framework, which has been pre-trained on 9M parallel sentences with syntheti-

[1] https://github.com/makcedward/nlpaug

[2] https://github.com/wolfgarbe/symspell

cally generated grammatical errors (Awasthi et al., 2019). We use the ensembled predictor that uses Sequence tagging, token-level transformations and two-stage fine-tuning with BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) pre-trained language models. This model is the current state-of-the-art on CoNLL-2014 Shared Task dataset (Ng et al., 2014) and BEA-2019 shared task dataset (Bryant et al., 2019).

▷ **NQE** (Chollampatt and Ng, 2018) : We use a Neural Quality Estimator with a Convolutional Neural Network (CNN) backbone. NQE provides a quality vector associated with the GEC's output which are fed in as a feedback feature to improve the GEC's performance.

▷ **BERT** (Devlin et al., 2019): We adapt the pre-trained BERT language model for the generative tasks. For the experiments we use the BERT-based model trained on BooksCorpus (Zhu et al., 2015) and English Wikipedia data. We use a tied Seq2Seq model with BERT as both encoder and decoder, followed by a dense layer for generation.

▷ **BART** (Lewis et al., 2020) : We use the denoising autoencoding pre-trained Seq2Seq language model BART as one of our baselines. We use the BART-base architecture (number of layers $L = 6$) in this study.

▷ **T5** (Raffel et al., 2020) : We fine-tune T5-base model, a transformer based text-to-text model pre-trained in a multi-task transfer learning setting. Similar to BART, T5 is also pre-trained with fill-in-the-blank-style of denoising objective.

We obtain the pre-trained model weights for BERT, BART and T5 from Huggingface's transformer library.[3]. Owing to space constraints, we report the different hyperparameters and the system settings in the appendix.

4.3 Results

4.3.1 Intrinsic Evaluation Results

We compute WRR and BLEU scores for evaluating models on intrinsic denoising experiment and report the results in Table 3. For OCR generated noises, we observe that pre-trained transformer-based Seq2Seq models attain better scores than other baselines, with T5-base performing the best with an average lift of 1.7% WRR and 4.8% BLEU on ALTA and an average lift of 4.0% WRR and BLEU on ICDAR, against the best performing spelling and GEC baseline (*i.e.,* GECToR). Further, it worth noting that the gated versions (for each gate) of all said transformer sequence models outperform their non-gated counterparts by 0.5% WRR and 2.5% BLEU.

However, the results for synthetically infused noises are far more significant. Here we observe a wider margin between transformer based correction models and the non-transformer models, NQE and Symspell. This shows the shortcomings of the confusion set-based spell correction methods on adversarial attacks. On the other hand, pre-trained language models are extremely accurate in correcting infused noises with 93.6% WRR and 94.4% BLEU, albeit having more noisy inputs (Table 1). Among all the models, T5 performs the best even on the infused noises. This is the case with the OCR datasets, that is, gated versions (each gate) of all said transformer sequence models outperform their non-gated counterparts by over 3.5% in terms of both WRR and BLEU scores. The margin is much wider between `gated-BERT` and BERT, which shows the weakness of the inbuilt denoising capability of BERT model, as compared to BART and T5. The superior performance of `gated-Trans` is indicative of the effectiveness of our model while dealing with noises generated by OCR systems as well as, adversarial attacks and man-made errors.

`gated-Trans` allows the flexibility to select the gates in the Seq2Seq model during encoding and generation. As part of ablation study we also report the results for different combinations of gates with the transformer models, namely *(i) mask+generate (MG)*, *(ii) copy+generate (CG)*, and *(iii) mask+copy+generate (MCG)*. `gated-Trans` with MG loosely resembles the Soft-Masked BERT model (Zhang et al., 2020). We observe that across different intrinsic datasets, `gated-Trans` with just CG or, MCG gates consistently outperform the all-gated version with a margin of over 0.5%. This points favorably to our hypothesis that, since denoising is a one-to-one sequential task, the skip gate is of lesser importance than the other three gates. Hence, exclusion of the skip gate boosts the performance of our models significantly. Additionally, we observe that under *Random* noise, `gated-Trans` with all gates performs significantly better than the other variants. In these cases, we observe a pivotal role by the *skip*

[3] https://huggingface.co/models

Models	Datasets											
	OCR Noise				Infused Noises (IMDb dataset)							
	ALTA		ICDAR		Random		Keyboard		Swap		Delete	
	WRR	BLEU	WRR	BLEU	WRR	BLEU	WRR	BLEU	WRR	BLEU	WRR	BLEU
Symspell	0.637	0.593	0.677	0.677	0.788	0.675	0.784	0.767	0.774	0.781	0.787	0.787
GECToR	0.823	0.778	0.769	0.774	0.899	0.864	0.971	0.970	0.955	0.963	0.950	0.953
NQE	0.753	0.729	0.756	0.714	0.809	0.786	0.872	0.855	0.842	0.829	0.850	0.808
BERT	0.818	0.754	0.765	0.765	0.872	0.872	0.964	0.982	0.842	0.926	0.831	0.899
g-BERT (*all gates*)	0.809	0.787	0.785	0.795	0.974	**0.984**	0.970	0.980	0.970	0.975	0.959	0.970
mask+generate	0.769	0.717	0.767	0.739	0.931	0.955	0.967	0.972	0.964	0.959	0.948	0.957
copy+generate	0.831	0.825	0.786	0.804	0.958	0.981	0.967	0.984	0.971	0.979	0.955	0.964
mask+copy+generate	0.825	0.819	0.798	0.805	0.969	0.975	0.973	0.986	0.977	0.977	0.971	0.975
BART	0.838	0.788	0.785	0.773	0.957	0.942	0.940	0.923	0.951	0.942	0.947	0.935
g-BART (*all gates*)	0.840	0.833	0.815	**0.824**	0.975	0.983	0.979	0.988	0.977	0.982	0.964	0.978
mask+generate	0.822	0.817	0.796	0.802	0.954	0.973	0.972	0.984	0.962	0.969	0.940	0.963
copy+generate	0.842	0.831	0.804	0.811	0.971	0.981	0.979	0.987	**0.983**	0.985	0.971	0.975
mask+copy+generate	0.844	**0.835**	0.814	0.820	0.972	0.983	0.980	0.984	0.973	0.978	0.976	0.988
T5	0.840	0.826	0.809	0.814	0.971	0.983	0.988	0.991	0.974	0.986	0.964	0.977
g-T5 (*all gates*)	0.825	0.808	0.814	0.823	**0.981**	0.983	0.991	**0.995**	0.978	**0.987**	0.978	**0.991**
mask+generate	0.827	0.807	0.793	0.799	0.955	0.976	0.974	0.983	0.977	0.977	0.937	0.955
copy+generate	0.841	0.826	0.812	0.823	0.974	0.982	0.988	0.990	0.980	0.978	0.978	0.989
mask+copy+generate	**0.849**	0.831	**0.823**	0.831	0.975	0.980	**0.992**	0.994	0.980	0.982	**0.984**	0.989

Table 3: Intrinsic Evaluations Across Baselines for Word Recognition Task. We highlight the best scores in bold.

gate, which detects insertion based noises and skips contextually irrelevant tokens to generate the clean target text.

4.3.2 Extrinsic Evaluation Results

For extrinsic evaluations we report BLEU and ROUGE scores for all the models for MT and summarization tasks with *Random*, *Keyboard*, *Swap* and *Delete* noises infused in the source text. We evaluate the performances of `gated-Trans` for both these tasks to determine the impact of effectively dealing with synthetic noises on the performances of downstream tasks.

Machine Translation: At an overall level, we observe in Table 4 that `gated-Trans` performs better than its non-gated counterparts w.r.t both BLEU and ROGUE scores, irrespective of type of infused noise. Similar to intrinsic evaluation, T5 outperforms the other transformer models for all noises. For *Random* noise, we observe that the `gated-BART` supersedes all other baselines, with a lift of 0.4% BLEU and 0.8% ROUGE over the most competitive benchmark of T5. Similarly, for *Swap* and *Delete* noises, `gated-T5` performs significantly better than all other models by a margin of 1.5% BLEU and 0.9% ROUGE. On the other hand, for *Keyboard* noise `gated-BART` with just MCG gates shows the best results with an improvement of 0.3% BLEU and 1.0% ROUGE over the T5-base model.

Summarization: For *Random*, *Keyboard* and *Delete* noises, `gated-T5` outperforms all other models with a significant margin of more than 0.5%. Also, for *Swap* noise, the `gated-T5` with MCG gates reports the best result across all variants.

An interesting conclusion from the extrinsic evaluation results is the effectiveness of all the gates in `gated-Trans` for generative tasks. This can be attributed to the fact that using *copy* and *generate* gates along with *skip* gate and these tasks being generative in nature often require tokens to be skipped in order to translate or summarize. Hence, while each gated version performs better than the base model, the *skip* gate provides an added advantage for generative tasks.

5 Result Analysis

In this section we analyze the performance of `gated-Trans` both quantitatively and qualitatively. We perform a statistical test of significance to statistically validate the effectiveness of gating mechanism in transformer based Seq2Seq models. We conduct one-tailed Welch's $t-$test (WELCH, 1947) on the intrinsic and extrinsic results of each pre-trained transformer model and the gated versions and reject the null hypothesis with a $p-value$ of 0.01. This indicates that the improvement by our gated model is not random and is attributed by the architectural novelty. Next we establish the relationships between gated-conditional probabilities and the volume of noise and overall length of the text. In Figure 2a, we present the linear relationship between the WRR score and the gating probability values. One can notice that the *masking* probability

Models	Infused Noises (WMT14)								Infused Noises (CNN/DM)							
	Random		Keyboard		Swap		Delete		Random		Keyboard		Swap		Delete	
	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE	BLEU	ROGUE
BERT	0.449	0.101	0.458	0.096	0.449	0.102	0.450	0.098	0.263	0.565	0.262	0.563	0.261	0.565	0.261	0.561
g-BERT (*all gates*)	0.454	0.096	0.465	0.096	0.459	0.093	0.466	0.094	0.265	0.568	0.268	0.569	0.262	0.568	0.264	0.564
mask+generate	0.452	0.099	0.461	0.095	0.451	0.091	0.455	0.091	0.262	0.566	0.263	0.565	0.260	0.561	0.263	0.562
copy+generate	0.453	0.099	0.460	0.089	0.453	0.094	0.459	0.101	0.261	0.567	0.264	0.566	0.262	0.567	0.260	0.559
mask+copy+generate	0.454	0.104	0.463	0.095	0.455	0.093	0.462	0.100	0.264	0.566	0.264	0.566	0.262	0.567	0.261	0.562
BART	0.472	0.142	0.469	0.121	0.470	0.115	0.474	0.109	0.300	0.589	0.309	0.595	0.298	0.576	0.289	0.569
g-BART *(all gates)*	**0.481**	**0.153**	0.471	0.114	0.474	0.118	0.479	0.119	0.302	0.589	0.321	0.619	0.308	0.594	0.302	0.614
mask+generate	0.477	0.146	0.470	0.116	0.470	0.117	0.475	0.111	0.303	0.590	0.312	0.611	0.314	0.597	0.298	0.586
copy+generate	0.477	0.146	0.471	0.114	0.472	0.116	0.477	0.110	0.301	0.589	0.312	0.606	0.302	0.585	0.296	0.581
mask+copy+generate	0.479	0.149	**0.477**	**0.130**	0.473	0.117	0.478	0.110	0.304	0.590	0.315	0.615	0.308	0.594	0.298	0.585
T5	0.477	0.145	0.474	0.120	0.483	0.128	0.490	0.119	0.324	0.640	0.329	0.655	0.313	0.638	0.323	0.637
g-T5 *(all gates)*	0.477	0.146	0.475	0.123	**0.494**	**0.134**	**0.508**	**0.131**	**0.329**	**0.649**	**0.334**	**0.661**	0.319	0.643	**0.325**	**0.640**
mask+generate	0.474	0.145	0.472	0.121	0.484	0.128	0.492	0.124	0.319	0.641	0.324	0.649	0.301	0.596	0.315	0.609
copy+generate	0.475	0.145	0.472	0.121	0.487	0.129	0.496	0.127	0.326	0.645	0.328	0.659	0.317	0.640	0.324	**0.640**
mask+copy+generate	0.479	0.150	0.476	0.127	0.491	0.132	0.503	0.129	0.326	0.646	0.329	**0.661**	**0.321**	**0.651**	0.323	0.638

Table 4: Extrinsic Evaluation on WMT14 English-French translation and CNN/DM summarization tasks

Input	Sys	**Prediction**
Source: *Girl Cricketer Recovers Having recovered from a head in- <*> jury, caused by a cricket ball during an interstate women's match in Ade- <*> laide on Saturday, Nesta Moon, 17, wicketkeeper of the Victorian team, was discharged from the the Adelaide &>?pltal after detention for observation. '*	A	*girl cricketer recovers having recovered from a head **injury**, caused by a cricket ball during an interstate women's match in **adelaide** on saturday, nesta moon, 17, wicketkeeperkeeper of the victorian team, was discharged from the adelaide **hospital** after detention for observation.*
Target: *Girl Cricketer Recovers Having recovered from a head **injury**, caused by a cricket ball during an interstate women's match in **Adelaide** on Saturday, Nesta Moon, 17, wicket-keeper of the Victorian team, was discharged from the Adelaide **Hospital** after detention for observation.*	B	*girl girl recovers having recovered from a head in jury caused by a cricket ball during an interstate womens match in **adelaide** on saturday night snow 17 wicketkeeper of the victorian team was discharged from the adelaide police after detention for observation*

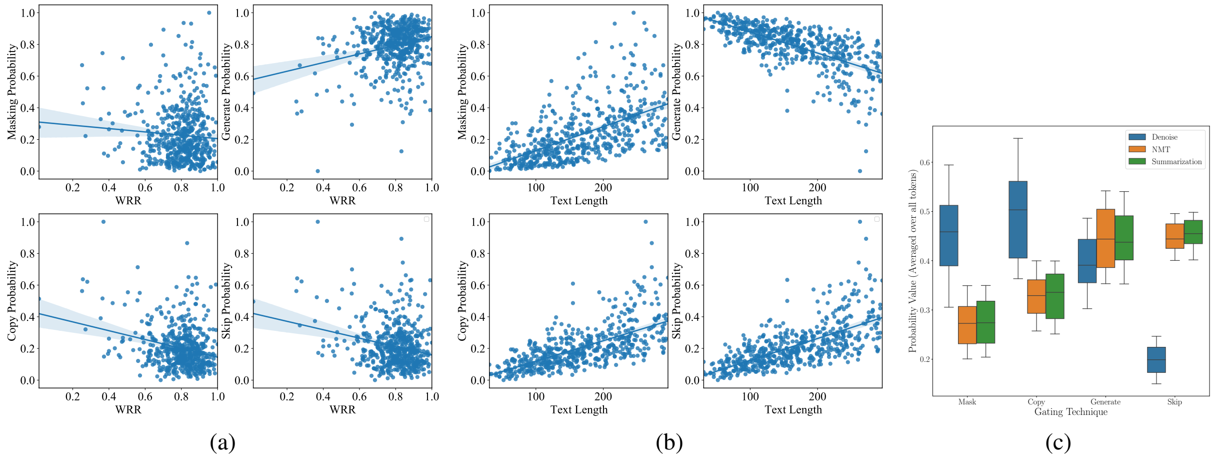
Table 5: Error Analysis on a sample instance on denoising task. System A denotes gated-T5 and B denotes T5-base. Tokens denoted by <*> indicate noises in the source text.

has a positive correlation with the noise amount (+0.08 correlation *w.r.t.* WRR). This is highly intuitive, since the *masking* gate should ideally *mask* the noisy tokens only. Also, in Figure 2b, it can be seen that only *generate* probability has a negative correlation (-0.66) with the length of text, while *masking, copy and skip* probabilities show positive trend with the text length. This might be owing to the fact that for shorter text, there is lack of context. To overcome this, the *generate* gate in gated-Trans will come in play to add more characters/words to the text to create a more contextually coherent text, that will naturally be devoid of noises. From 2c we can further strengthen the hypotheses on relative importances of each of the gates. For one-to-one translation task like denoising, we can observe a higher *masking* and *copy* probability. On the other hand, in generative tasks like MT and summarization tasks, *generate* and *skip* gates play more vital roles with higher probability to generate more meaningful target.

Lastly, we take a look at few examples to interpret the superior performance of gated-Trans . In Table 6b, we showcase an example from denoising task and compare gated-T5 with T5-base. We provide few more instances in the appendix. We also report the *copy* and *masking* probability heatmaps in Figure 3. In the provided example, the source text contains three noises occurring at *injury*, *Adelaide* and *Hospital*. While gated-T5 is able to rectify all the three noises, T5 base model is able to correct only one (*Adelaide*). We can also observe in the heatmaps in Figure 3a and 3b, that all the noisy characters and tokens have high masking probabilities, which is consistent with the expected behavior of the *masking* gate in gated-Trans .

6 Conclusion

In this work, we propose gated-Trans - a novel end-to-end Seq2Seq Transformer model with conditional gates for robust generation. The gated-unit in our sequential model is efficient for effectively removing noises from texts which has been demonstrated by its superior performance over other competitive baselines. We also observe the performance improvement that gated-Trans is achieved in downstream tasks like machine translation and summarization. Further, we showcase the noise-invariant nature of gated-Trans , which is capable of removing not only OCR induced noises but also synthetically infused noises, highlighting its potential efficacy towards dealing with

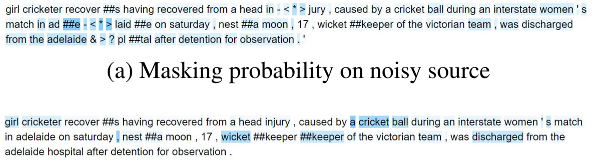


(a)

(b)

(c)

Figure 2: Average masking, copy, generate and, skip probabilities for different (a) Noise levels; (b) Text lengths (c) Tasks



girl cricketer recover ##s having recovered from a head in - < * > jury , caused by a cricket ball during an interstate women ' s match in ad ##e - < * > laid ##e on saturday , nest ##a moon , 17 , wicket ##keeper of the victorian team , was discharged from the adelaide & > ? pl ##tal after detention for observation . '

(a) Masking probability on noisy source

girl cricketer recover ##s having recovered from a head injury , caused by a cricket ball during an interstate women ' s match in adelaide on saturday , nest ##a moon , 17 , wicket ##keeper ##keeper of the victorian team , was discharged from the adelaide hospital after detention for observation .

(b) Copy probability on a sample target

Figure 3: We highlight high masking and copy probability for different word tokens for a denoise instance

a range of adversarial attacks on texts. Further, our model allows more flexible text generation suitable for different purposes just by opting different gates out. Looking forward, we would like to explore and handle noises which specifically occur in OCR and speech-to-text conversion processes by leveraging information from multiple modalities.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364.

Youssef Bassil and Mohammad Alwani. 2012. Post-editing error correction algorithm for speech recognition using bing spelling suggestion. *International Journal of Advanced Computer Science and Applications*, 3(2).

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 247–253.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

G. Chiron, A. Doucet, M. Coustaty, and J. Moreux. 2017. Icdar2017 competition on post-ocr text correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1423–1428.

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016. Adapting grammatical error correction based on the native language of writers with neural network joint models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP'16*, pages 1901–1911. The Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 2528–2539, Brussels, Belgium. Association for Computational Linguistics.

Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. 2010. Unsupervised cleansing of noisy text. In *Coling 2010: Posters*, pages 189–196. Coling 2010 Organizing Committee.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT'19, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152.

Jianfeng Gao, Chris Quirk, et al. 2010. A large scale ranker-based system for search query spelling correction.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. pages 140–149.

Jinxi Guo, Tara N Sainath, and Ron J Weiss. 2019. A spelling correction model for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5651–5655. IEEE.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, pages 1693–1701. Curran Associates, Inc.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting N-Best Hypotheses to Improve an SMT Approach to Grammatical Error Correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16*, pages 2803–2809. IJCAI/AAAI Press.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. Faspell: A fast, adaptable, simple, powerful chinese spell checker based on daedecoder paradigm. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 160–169.

Kenji Imamura, Eiichiro Sumita, and Yuji Matsumoto. 2003. Automatic construction of machine translation knowledge using translation literalness. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556. Association for Computational Linguistics.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL'20*, pages 4248–4254. Association for Computational Linguistics.

Vlado Keselj. 2009. Speech and language processing daniel jurafsky and james h. martin (stanford university and university of colorado at boulder) pearson prentice hall, 2009, xxxi+ 988 pp; hardbound, isbn 978-0-13-187321-6, $115.00.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In *Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB'05, Proceedings*, volume 3513 of *Lecture Notes in Computer Science*, pages 263–274. Springer.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR'15, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Amrith Krishna, Bodhisattwa P. Majumder, Rajesh Bhat, and Pawan Goyal. 2018a. Upcycle your OCR: Reusing OCRs for post-OCR text correction in Romanised Sanskrit. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 345–355, Brussels, Belgium. Association for Computational Linguistics.

Amrith Krishna, Bodhisattwa Prasad Majumder, Rajesh Shreedhar Bhat, and Pawan Goyal. 2018b. Upcycle your OCR: reusing ocrs for post-ocr text correction in romanised sanskrit. In *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL'18*, pages 345–355. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL'20*, pages 7871–7880. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Eranga Mapa, Lasitha Wattaladeniya, Chiran Chathuranga, Samith Dassanayake, ND Silva, Upali Kohomban, and Danaja Maldeniya. 2012. Text normalization in social media by using spell correction and dictionary based approach. *Systems learning*, 1:1–6.

Diego Molla and Steve Cassidy. 2017. Overview of the 2017 alta shared task: Correcting ocr errors. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 115–118.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Stephen Mutuvi, Antoine Doucet, Moses Odeo, and Adam Jatowt. 2018. Evaluating the impact of ocr errors on topic modeling. In *International Conference on Asian Digital Libraries*, pages 3–14. Springer.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Elvys Linhares Pontes, Ahmed Hamdi, Nicolas Sidere, and Antoine Doucet. 2019. Impact of ocr quality on named entity linking. In *International Conference on Asian Digital Libraries*, pages 102–115. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

C. Rigaud, A. Doucet, M. Coustaty, and J. Moreux. 2019. Icdar 2019 competition on post-ocr text correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Felix Stahlberg, Christopher Bryant, and Bill Byrne. 2019. Neural grammatical error correction with finite state transducers. 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 4033–4039, Minneapolis, Minnesota. Association for Computational Linguistics.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785.

B. L. WELCH. 1947. THE GENERALIZATION OF 'STUDENT'S' PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARLANCES ARE INVOLVED. *Biometrika*, 34(1-2):28–35.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling Error Correction with Soft-Masked BERT. pages 882–890.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Dataset

A.1 Noises in Texts for Intrinsic Evaluation

The intrinsic experimental datasets contain OCR extracted errors which comprise of variations of spelling errors. These errors can be categorized on different ways which are as follows: -

• Depending on edit distance, there are single-error tokens with edit distance of 1 (e.g. ‘school’ vs. ‘schopl’) and multi-error tokens with higher edit distances (e.g. ‘school’ vs. ‘schopi’).

• Misspellings can occur at the first character (e.g. ‘world’ vs. ‘uorld’) or at other characters (e.g. ‘world’ vs. ‘workd’, ‘world’ vs. ‘worlh’). The average rate of first-position errors can then be considered to be around 11% of misspellings. Further, a non-word error is when a token is not a lexicon entry and real-word error is when a valid word occurs in a wrong context. For example, in two phrases ‘glow-worm candles’ and ‘glow-wonn candies’, a non-word error is ‘glow-wonn’ while ‘candies’ is a real-word error.

• In case of problems with word boundaries, wrongly deleting/inserting white spaces results in run-on errors (e.g. ‘is said’ vs. ‘issaid’).

A.2 Infused Noises

Below we describe the different types of noise injection techniques we explore in this work. We infuse these set of noises artificially to the source texts of IMDb, MT and summarization datasets.

▷ **Random augmentation:** This type of augmentation occurs when one of the characters from a token is randomly inserted, deleted or substituted.

▷ **Keyboard augmentation:** This type of augmentation is the result of keyboard error of a token where one of the character is replaced by its neighbouring character in the keyboard position. example could be ‘jumps’ vs ‘juJps’.

▷ **Swap augmentation:** This sort of augmentation occurs at both character level as well as word level. In Character level augmentation, characters of a particular word are swapped within the same word. However at word level augmentation, two or more words within a sentence are swapped. In other words, the positions of words in a sentence are changed. This type of augmentation method changes the linguistic meaning of a sentence.

▷ **Delete augmentation :** As the name suggests, under this augmentation method, one of the characters across every token or word of the sentences is deleted. Missing or deleted character could be anywhere from the token of a sentence. Example of Delete Augmentation is: Original Text: *“The quick brown jumps over.”* Augmented Text: *“Te quic rown fx jump ver.”*

B Experiment Setup and Hyperparameters

For all the models across each of the datasets we use 300 as the maximum token length of a source text as well as the target text. We resort to padding technique to maintain fixed source length. However for MT, we use the maximum length as 100. For all the models we use categorical cross entropy to calculate the loss. Transformer models are trained with Adam ($\eta = 5e-5, \beta_1 = 0.9, \beta_2 = 0.999$) optimizer (Kingma and Ba, 2015) with a weight decay rate of 0.001. For GECToR and NQE we use Adam optimizer with $\eta = 1e-3$ without any weight decay. We train our models for 30 epochs with an early-stopping criteria ($patience = 10$) based on the validation loss. All models are trained with *batch-size* of 32. As none of the datasets contain separate train-dev split, we split each datasets into 80-10-10 for training-validation and testing. We conduct all our experiments on a single Tesla T4 GPU. Average runtime to train and validate a single batch of size 32 is 4.2 seconds and 1.5 seconds respectively. We primarily use BERT, BART and T5 pre-trained language models as the backend for both encoder and decoder in this work. Due to this transfer learning setup, our model requires a minimal task-specific dataset to work with. Further, we train our model with different learning rates in different layers (smaller learning rate in language model backend and larger in task-specific layers), making it pretty effective even on smaller datasets. Size-wise gated-transformers are comparable with their non-gated counterparts. *e.g.* BART model has 139M parameters. Compared to that, a gated-transformer with a BART backend has only 3076 additional parameters.

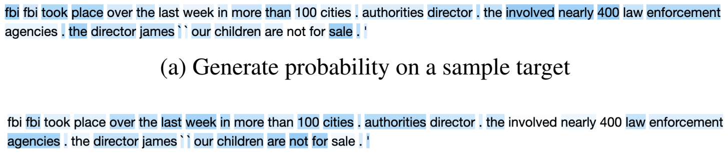
Input	Sys	**Prediction**
Source: *The FBI hase rescued 168 children and arristed 281 pimps in a countrywide crackdown on child sex trafficking. The operation, which took place over the last week in more than 100 cities, involved nearly 400 law enforcement agencies, authorities said Monday. The message, said FBI Director James Comey, should be clear: ‘ ‘ Our children are not for sale. .. . We will respond and crush these pimps who would crush these children. ’ ’*	A	*fbi fbi took place over the last week in more than 100 cities. authorities director. the involved nearly 400 law enforcement agencies. the director james ‘ ‘ our children are not for sale.’*
Target: *The operation took place over the last week in more than 100 cities , FBI says.It involved nearly 400 law enforcement agencies.FBI director : “ Our children are not for sale ”*	B	*the fbi involved place over the last week in more than 100 cities. authorities director. 168 involved 281 400 law enforcement agencies. the director : ‘ ‘ our children are not for sale.*

(a) Summarization

Input	Sys	**Prediction**
Source: *Atomic Chief Sir William Penney, JJrilain’s chief atomic scientist, on his arrival at i’tiraneld yesterday.*	A	*atomic chief sir william penney, chief’s chief atomic scientist, on his arrival at ti graph yesterday, sir*
Target: *Atomic Chief Sir William Penney, **Britain’s** chief atomic scientist, on his arrival at **Parafield** yesterday.*	B	*atomic atomic scientist sir william penney july 1 sir william penney john william penney jjrilains chief atomic scientist on his arrival at itiraneld yesterday*

(b) Denoise

Table 6: Error Analysis on sample instances. System A denotes `gated-T5` and B denotes T5-base model.



fbi fbi took place over the last week in more than 100 cities . authorities director . the involved nearly 400 law enforcement agencies . the director james ` ` our children are not for sale . '

(a) Generate probability on a sample target

fbi fbi took place over the last week in more than 100 cities . authorities director . the involved nearly 400 law enforcement agencies . the director james ` ` our children are not for sale . '

(b) Copy probability on a sample target

Figure 4: Heatmap of token-wise probability values for sample source and targets.

C Error Analysis

Looking to the summarization example in Table 6a, we observe once again `gated-T5` yields a much cleaner text compared to T5-base model. T5 ends up keeping random numbers in the predicted text, whereas the gated model is able to meticulously clear out the contextually insignificant tokens from the text before summarization. This is indicative of the contextual semantic understanding that our gating mechanism showcases in our model. Also, we observe that in the beginning of the predicted summarized text, `gated-Trans` ends up predicting two *fbi*’s. This can be attributed to the fact that the first token has no context to depend on, hence the *copy* probability in Figure 4 can be seen as low for the first predicted token and high for the second token, while the *generate* probabilities follow the inverse trend. Further, in the denoising example in Table 6b, it can be observed that both `gated-T5` and T5 base model fail to effectively denoise the source text. However it is worth noting that `gated-T5` outputs are much cleaner (albeit imperfect) text compared to T5, which ends up injecting more noise into the text.