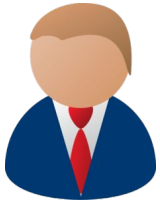
Planning and Generating Natural and Diverse **Disfluent Texts** as Augmentation for **Disfluency Detection**

Jingfeng Yang, Zhaoran Ma, Diyi Yang

Georgia Tech College of Computing
School of Interactive Computing

Disfluency Detection and Disfluency Generation

I want a flight **to Boston um** to Denver



Disfluency Detection

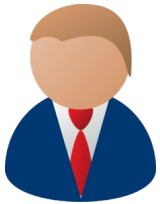
I want a flight to Denver

First step of processing transcribed speech, which is the preliminary of other downstream NLP tasks.

It's nothing but wood down here

Disfluency Generation

It's nothing but wood **up here** down here



Help text-to-speech systems to generate more natural speech.

Help disfluency detection.

Prior Work on Disfluency Detection

- Noisy Channel Models (requires Tree Adjoining Grammar transducer)
  Zwarts and Johnson, 2011; Lou and Johnson, 2018
- Transition based models (often requires phrase structure)
  Rasooli and Tetreault, 2013; Yoshikawa et al., 2016; Wu et al., 2015; Jamshid Lou and Johnson, 2020
- Sequence-tagging models (requires no ad-hoc features recently and achieves SOTA)
  Ferguson et al., 2015; Hough and Schlangen, 2015; Zayats et al., 2016; Lou and Johnson, 2018; Wang et al., 2019
- End-to-end generation models
  Wang et al., 2016, 2018

Using augmented disfluent data in disfluency detection

Problem: lack of annotated disfluent data to train disfluency detection models

Step 1	Step 2	Step 3
Generate fake disfluent texts with disfluent segment labels as augmented data	Use augmented disfluent data to pretrain the disfluency detection sequence labeling model	Fine-tune the disfluency detection model on gold labeled disfluency dataset

Quality of generated disfluent texts

Disfluency type:

Type	Example
Repetition	~~they~~ they learn to share.
Deletion	this ~~is~~ just happened yesterday.
Substitution	it's nothing but wood ~~up here~~ down here.

Problem: disfluent texts generated by random insertion or repetition of ngrams
1) are not natural 2) lack Substitution type of disfluency
Thus, they do not resemble gold disfluency data

METHODS	EXAMPLE
Random repetition	(1) ~~that 's~~ that 's really good.
Random insertion	(2) ~~of a~~ that 's really good.
Generation (ours)	(3) ~~it 's~~ that 's really good.
Generation (ours)	(4) that would ~~be more~~ be worth doing.

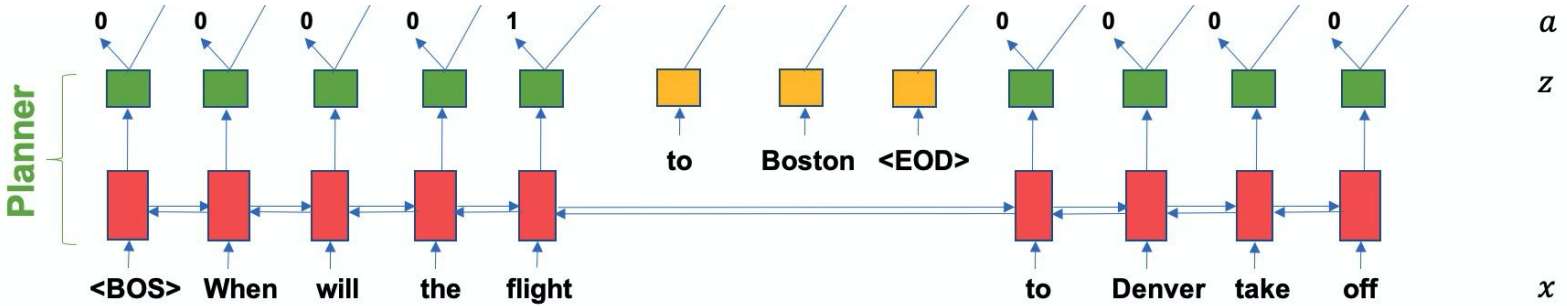
Generating natural & diverse augmented disfluent texts

Planner chooses which positions to insert disfluent segments.

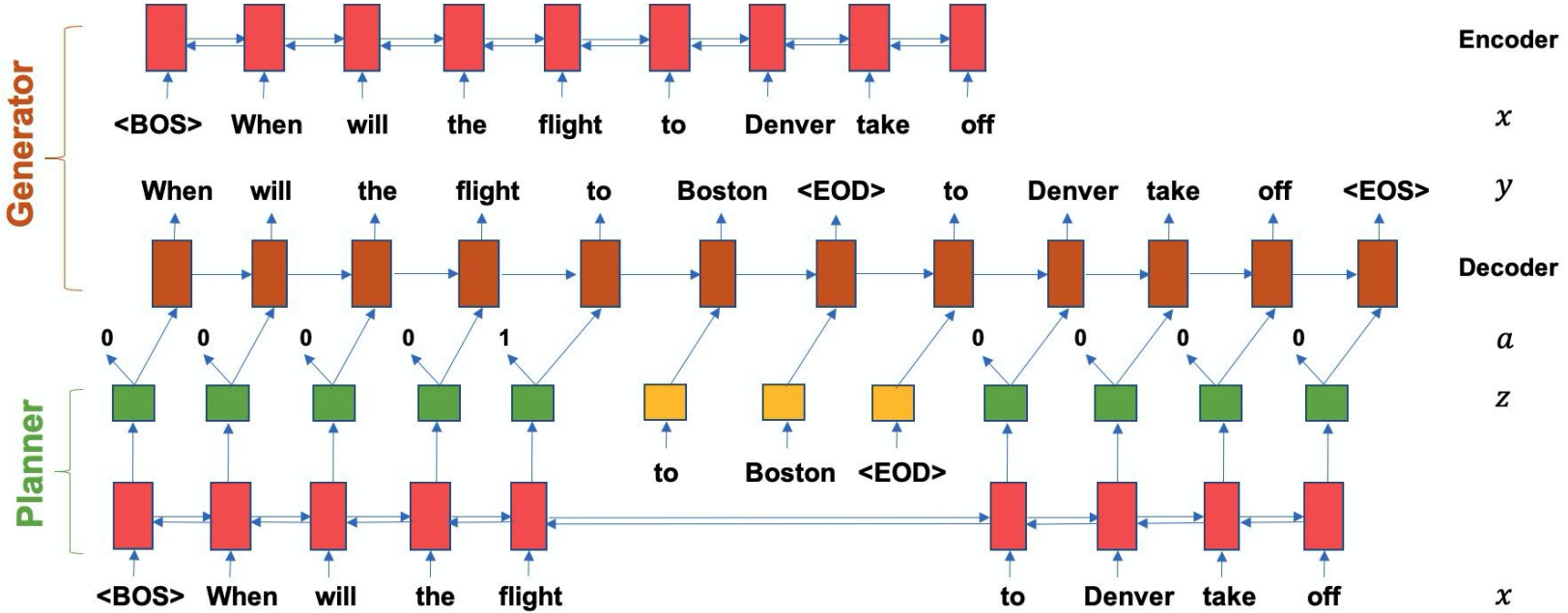
Generator learns and generates natural disfluent segments in picked positions.

Planner-Generator (PG) model

$$z_j = \begin{cases} h_i & \text{if } a_i = 0 \text{ or } (a_i = 1 \text{ and} \\ & y_j \text{ is the first word of reparandum)} \\ E(y_{j-1}) & \text{otherwise,} \end{cases}$$



Planner-Generator (PG) model



Dataset and Variants of Planner-Generator (PG) model

High Accuracy

High Diversity

PG

Full model

PG-NC

Without passing state from Planner to Generator

PG-NC-AD

Without Attention mechanism between Generator Encoder and Generator Decoder

PG-NC-AD-ID

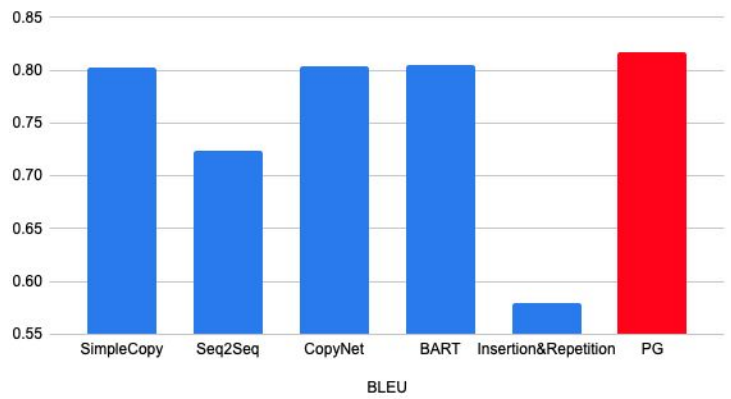
Without passing state from Generator Encoder to Generator Decoder (Generator degenerates to a LM)

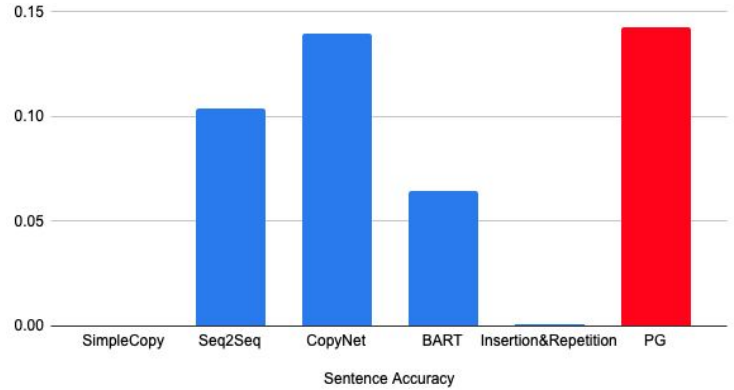
Disfluency Dataset: Switchboard Dataset (29k in 173k sentences are disfluent sentences)

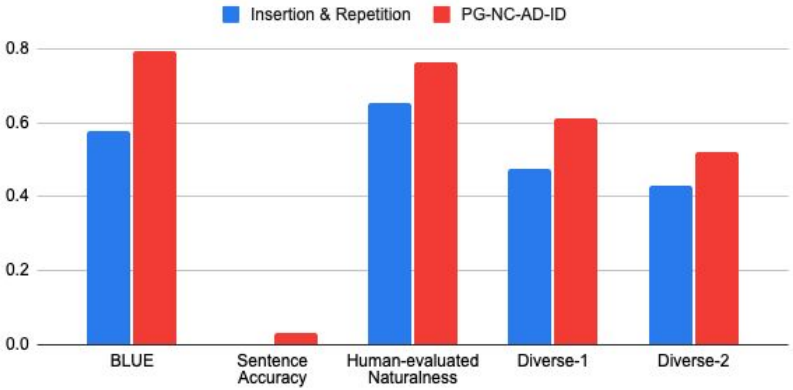
Unlabeled fluent data for augmentation: WMT 2017 monolingual dataset (3M sentences)

When doing augmentation, based on PG-NC-AD-ID, we replace LSTM Generator with a GPT2 Generator, and replace LSTM Planner with a heuristic Planner.

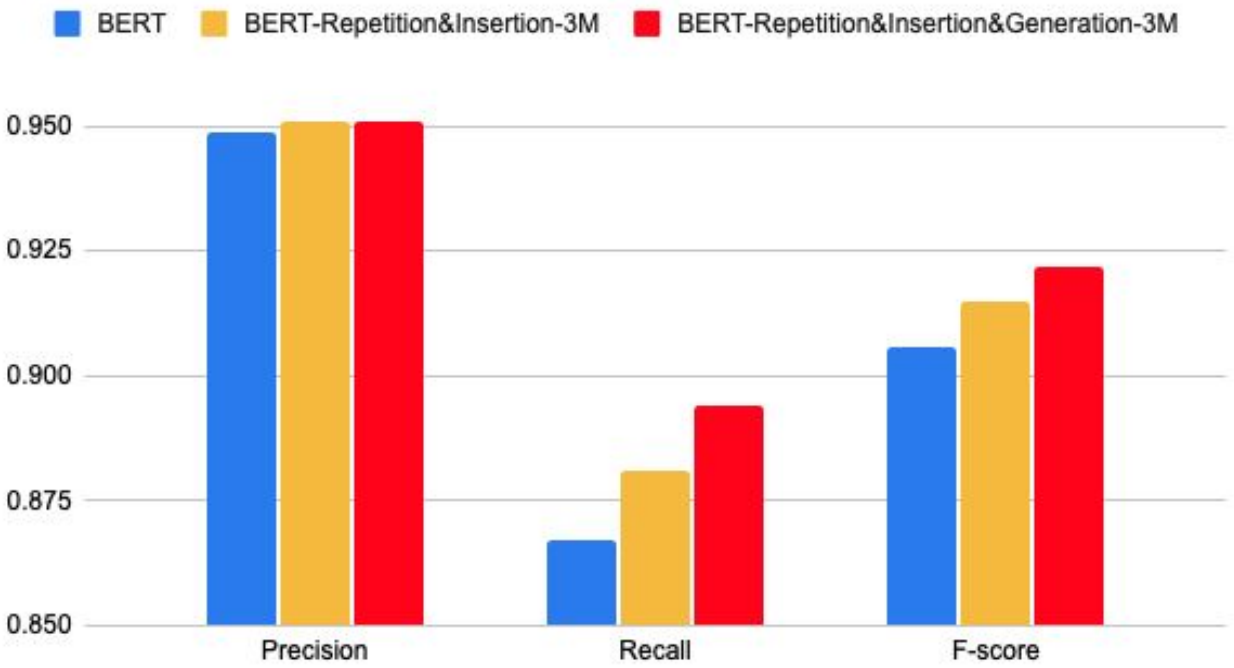
Disfluency Generation Results





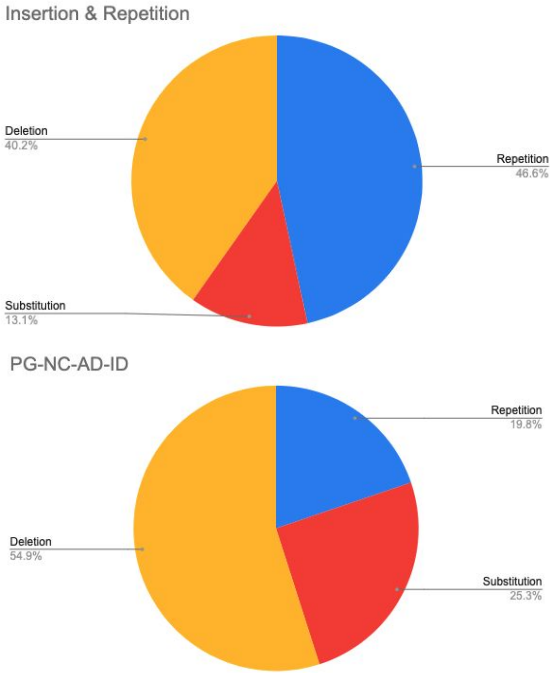


Disfluency Detection Results

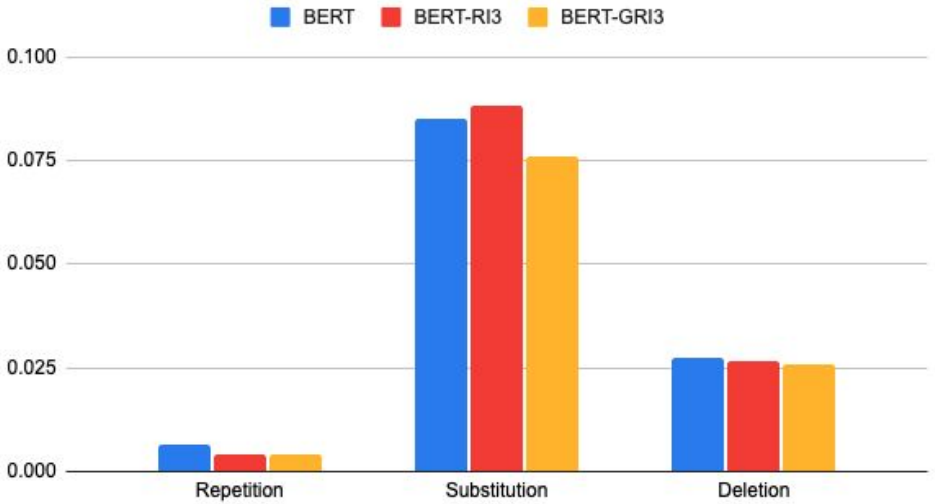


Results Analysis

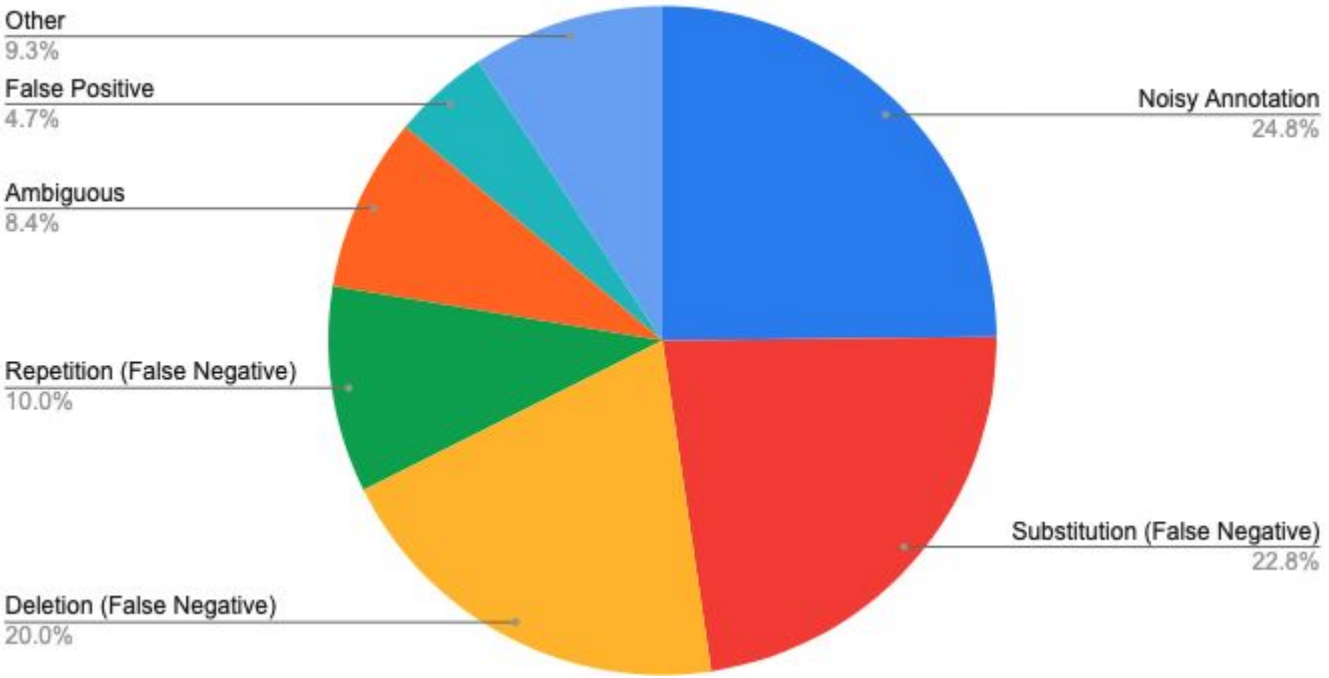
Generation Types:



False negative errors:



Error Analysis



Thank you!

Project Page:

Project Link:
https://github.com/GT-SALT/Disfluency-Generation-and-Detection