University of Bonn
CAISA lab

Introduction to Natural Language Processing

(Winter Semester 2023/2024)

Default Project

Project Report of Team #1

Formality Style Transformation

Group Members:

Duc Manh Vu	-	50137408
Suraj Giri	-	50190564
Muslimbek Abduvaliev	-	50136555
Nijat Sadikhov	-	50186266
Akmalkhon Khashimov	-	50178353

February 11, 2024

1 Introduction

In recent years, Natural Language Processing has witnessed advancements in various domains, including text generation, machine translation, and style transfer. Following our project proposal, we developed and compared various models for Machine Translation of informal sentences to their formal equivalents while preserving the meaning and intended message of the original text. The primary purpose of the project is to investigate various algorithms that can be used for this transformation.

In this project we sought to develop and evaluate multiple innovative approaches that can produce formal sentences and apply various metrics to check their accuracy and fluency. Our main goal was to find an answer to the question: "How do various models and algorithms for the formality style transformation compare in terms of their ability to effectively convert informal sentences into their formal equivalents?" In seeking for an answer to our questions, we evaluated the performance of four models: a General RNN model [1], a Bi-directional RNN model with LSTM [2], a pre-trained BERT model and its fine-tuned version [3], and a pre-trained CoEdit Large model and its fine-tuned version[4].

In this report, we have provided a comprehensive overview of the Formality Style Transformation task. We will start by taking a look at some of the previous research works done in this field. After that we will discuss about the dataset used and our data exploration process. Next, we will explore the various models we developed and compare their performances using various metrics. Lastly we will analyze the evaluation and results leading to the Analysis and Discussions and conclusions based on our findings.

2 Literature Review

There has been significant research conducted in the field of Formality Style Transfer. In this section, we will delve into some of the key studies conducted in this area. To start with, we will look into the work done by Sudha Roy and Joel Tetreault. In their work *Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer* [5], they provide a comprehensive overview of the dataset used by us for our project. Not only the dataset overview, they also provide insights and various resources for advancing the formality style transfer by offering a large corpus, strong benchmarks, and insightful discussions on automatic metrics and their limitations.

Similary, Ruochen Xu, Tao Ge, and Furu Wei propose a bi-directional framework augmented with hybrid annotations for formality style transfer in their work *Formality Style Transfer with Hybrid Textual Annotations* [6]. By leveraging a bidirectional style transformation seq2seq model, the authors fully exploit formality style classification data through classification feedback and various reconstruction constraints to facilitate model learning.

Correspondingly, Zhengyuan Liu and Nancy F. Chen's *Learning from Bootstrapping and Stepwise Reinforcement Reward: A Semi-Supervised Framework for Text Style Transfer* [7] introduces a novel framework that combines supervised and unsupervised approaches for text style transfer. Using automatically constructed pseudo parallel data and lexical- and semantic-based sentence matching techniques, the model achieves state-of-the-art performance across multiple datasets. Additionally, a stepwise reward re-weighting mechanism enhances generation performance, even

with minimal training data (10 % of the original size).

3 Data Exploration and Preprocessing

3.1 Dataset Information

Grammarly's Yahoo Answers Formality Corpus (GYAFC) [5] is the only dataset used in this project. This was created based on the Yahoo Answers L6 corpus[1] in two topics: *Entertainment & Music* (EM) and *Family & Relationships* (FR). Around 53,000 instances were collected on each topic to form the training set. Regarding the validation and test sets, the expert annotators created four different references for every sentence in each set. For each domain, they sampled around 3,000 instances for the validation set and 1,500 instances for the test set.

3.2 Exploratory Data Analysis

In this section, we will provide several statistics on the GYAFC dataset. Specifically, we will calculate the figures for each set and each topic. First, we will tokenize sentences using *en_core_web_sm* model from *spaCy* [8]. Then, we will use the lemma of all tokens to produce the statistics and the vocabulary. Singletons are lemmas that appear only once on the training set, while Out-Of-Vocabularys (OOVs) appear on the validation or test set but not in the training set. The meaningless lemmas do not have any meaning. We calculated the amount of meaningless words in the formal domain using the *WordNet* [9] corpus in the *Natural Language Toolkits* (NLTK) module [10]. The detailed statistics are shown in Table 1.

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=l

		Train		Val								Test							
		EM	FR	EM				FR				EM				FR			
				Ref. 0	Ref. 1	Ref. 2	Ref. 3	Ref. 0	Ref. 1	Ref. 2	Ref. 3	Ref. 0	Ref. 1	Ref. 2	Ref. 3	Ref. 0	Ref. 1	Ref. 2	Ref. 3
# sentences		52,595	51,967	2,877				2,788				1,416				1,332			
Informal	Avg. length	12.1407	12.6247	12.1175				12.2636				12.3213				12.7260			
	Vocabs	29,095	17,891	4,620				3,128				2,910				2,089			
	Singletons	16,515	10,075	-				-				-				-			
	OOVs	-	-	803				536				381				268			
Formal	Avg. length	12.6265	13.1134	12.6041	12.5613	12.5579	12.6232	12.3275	12.2131	12.2984	12.2220	12.7542	12.6999	12.7408	12.7338	12.9077	12.9317	13.1269	12.9775
	Vocabs	24,028	13,969	4,455	4,439	4,428	4,495	2,772	2,787	2,844	2,788	2,931	2,850	2,867	2,820	1,982	2,014	2,002	1,975
	Singletons	12,003	6,381	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	OOVs	-	-	527	525	530	545	262	272	261	264	286	269	254	251	137	154	152	148
	Meaningless	8,755	2,321	688	688	681	716	198	212	206	203	430	413	409	402	151	150	160	158

Table 1: Key statistics of the GYAFC dataset

For the volume of data, this dataset has a smaller amount compared to *WMT14 (French-English)* [11]. According to the authors, The *WMT14 (French-English)* [11] has 14.8 million instances in the training set, while there are around 105 thousand pairs in the GYAFC's training set.

3.3 Data Preprocessing and Augmentation

We implemented data preprocessing techniques for a dataset containing informal and formal sentences. The preprocessing tasks include text normalization, expansion of contractions, removal of repeated punctuations, and capitalization of the first letter of each sentence and proper nouns.

Preprocessing Steps:

Tokenization and POS Tagging: The NLTK library is used for tokenization and part-of-speech tagging. This helps in identifying proper nouns in the text.

Capitalization of First Letter and Proper Nouns: SpaCy is employed to identify the first letter of each sentence and proper nouns. Proper nouns and the first letter of each sentence are capitalized, while retaining the rest of the text as it is.

Expansion of Contractions: The contractions library is used to expand contractions in the text. For example, "don't" is expanded to "do not".

Removal of Repeated Punctuations: Regular expressions are used to remove consecutive repeated punctuations, ensuring text clarity and readability.

General Spacing Fixes: Regular expressions are applied to fix spacing issues such as space before punctuation, space after punctuation, and space in contractions. Multiple spaces between words are reduced to a single space.

Creating JSON File for Training Data:

- Informal and formal sentences are read line by line from corresponding files.
- Each line of informal text is preprocessed using the defined functions.
- The preprocessed informal text along with its corresponding formal text is appended to a list as a dictionary.
- A JSON file is generated containing the training data, where each data entry includes an ID, topic, and transformations.

Distribution of Informal and Formal Text Lengths: This histogram visualizes the distribution of text lengths (in terms of the number of words) for both informal and formal texts from the dataset. The x-axis represents the number of words in the texts, while the y-axis represents the frequency or count of texts having a particular length. This visualization helps in understanding potential differences or similarities in text lengths between informal and formal styles, which can provide insights into the characteristics of the dataset and guide further analysis or modeling decisions.

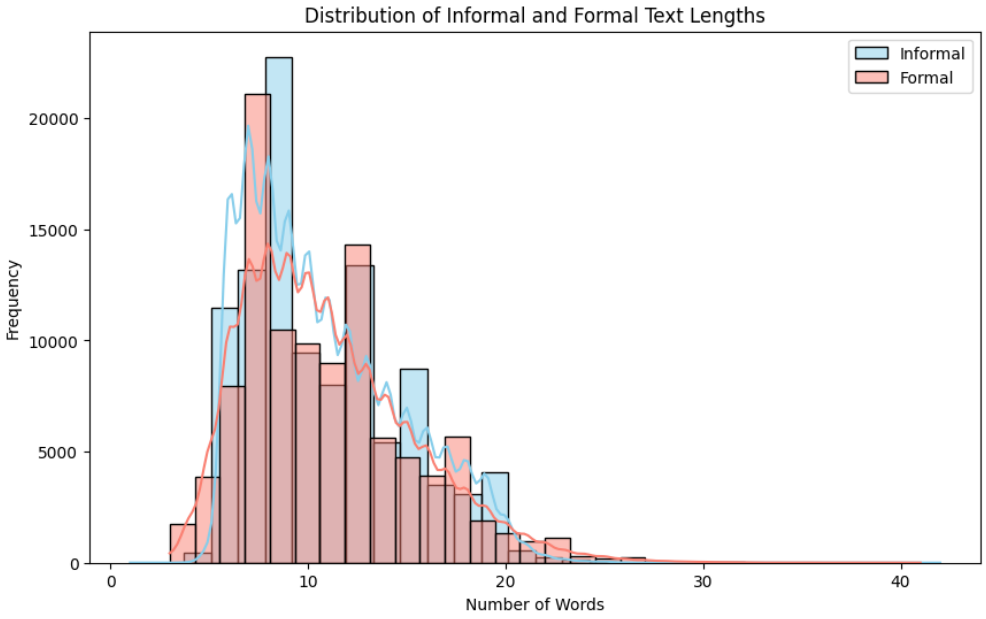

Figure 1: Distribution of Informal and Formal Text Lengths (Training data)

4 Methodology

4.1 RNN Model

We employ the RNN model built with PyTorch. The major aim of the project was to build a Reccurent Neural Network (RNN) model for Machine Translation of informal sentences to formal sentences. This implementation of the RNN included a straighforward interface, its internal architecture involves hidden layers and state transitions that process sequential data [12], as shown in Figure 2.

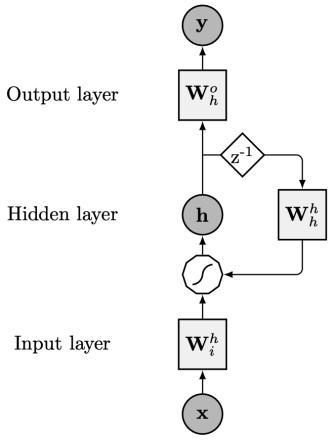

Figure 2: Simple RNN architecture [1]

4.2 BiRNN with LSTM

For this task, we employed the Bi-Directional Recurrent Neural Network (BiRNN) model with Long Short Term Memory (LSTM) using Pytorch. With the aim of Machine Translation of Formal sentences to Informal sentences, we used the GYAFC dataset provided by Grammerly. We used the Many-to-many architecture with Bi-LSTM using PyTorch, which achieved promising results, effectively transforming informal sentences to a more formal counterpart while preserving their original meaning. The many-to-many architecture can be seen in the Figure 3.

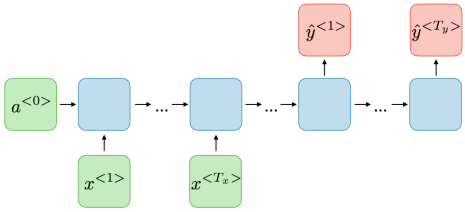

Figure 3: Many-to-many RNN architecture [2]

4.3 Pretrained BERT2BERT

We employ this pre-trained encoder-decoder architecture from Von Platen's work[2]. He trained this model on the CNN/DailyMail dataset [13] to solve the summarization task. The encoder and decoder are the Bidirectional Encoder Representations from Transformers (BERT) architectures [3]. BERT is just the encoder from the Transformers' architecture [14], which is shown in Figure 4.

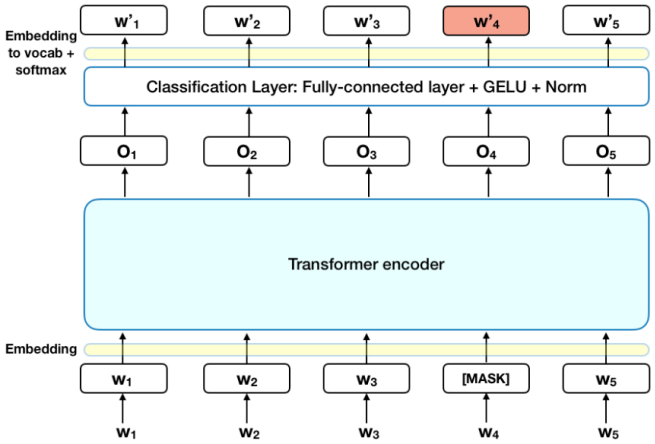

Figure 4: BERT architecture [3]

4.4 CoEdit

CoEdit [4] is the work from *Grammarly*. They introduce a new dataset for the non-meaning-changed editing task, which does not modify the information conveyed from the original sentence. The dataset includes prompts to instruct the purpose of the specific subtask: **Fluency**, **Coherence**, **Clarity**, **Paraphrasing**, **Formalization**, and **Neutralization**. They use this dataset to

[2] https://huggingface.co/patrickvonplaten/bert2bert-cnn_dailymail-fp16

		TERp	PINC(n=2)	BLEU	TER
General RNN model	val	0.126	0.561	0.406	0.440
	test	0.134	0.544	0.422	0.415
BiRNN with LSTM	val	0.127	0.558	0.412	0.437
	test	0.138	0.538	0.427	0.409

Table 2: Average performance of RNN models on the validation and test sets of GYAFC [5] dataset.

	Loss	
Epoch	General RNN model	BiRNN with LSTM
1	5.68937271651031	5.64649626018076
2	5.276047459138943	5.1782613426556345
3	5.112479813095989	4.963087262547002
4	5.0334168777617565	4.84183089801702
5	4.986323413533709	4.760151210948916
6	4.952393132440901	4.697540205870293
7	4.922186558585126	4.643745141995288
8	4.900054754796489	4.600115753081028
9	4.878925438318287	4.560939538289167
10	4.853318162034454	4.527787081906379

Table 3: Cross Entropy Loss of General RNN model and BiRNN with LSTM model for each epoch rained for 10 epochs.

finetune three different pre-trained FlanT5 [15] models, which are just pre-trained Transformers architectures [14] finetuned with instructive prefixes for multi-downstream tasks [16]. Regarding the Formalization, they finetuned the model with approximately 12,000 distilled instances from the GYAFC dataset.

5 Experimental Setup

In our work, we evaluate the performance of different architectures on both raw and preprocessed data versions. First, we train the RNN and LSTM models with these datasets. When we compared the results of the general RNN model with the results of a Bi-RNN model which uses LSTM as the recurrent layer. In the general RNN model, we used the general instance of the RNN as the recurrent layer. In both of the implementations of our RNN models, we firstly pre-processed the data using our Rule Based Pre-processing techniques. Since the models we developed using PyTorch are not the pre-trained models, we trained both the models for 10 epochs each. For both the RNN models, we used the instance of Embedding Layers as the input layer and corresponding recurrent layers and applied the linear transformation for the output layer. The results of the metrics can be seen in the Table 2

Similarly, the cross entropy loss for each epoch for each model is show in the Table 3

Regarding pre-trained large language models, we finetune the BERT2BERT model from Von Platen's work for 10 epochs. Then, we employed the large version of CoEdit for our work. For this architecture, we add a command prefix before each data instance and conduct experiments in two

directions. First, we validate the dataset with the original weights from the authors. Second, we finetune the model with both datasets for 5 epochs. Both BERT2BERT and CoEdit models and their pre-trained weights are available on Huggingface [17].

We use most of the preset protocol and configurations from Huggingface [17] to finetune both BERT2BERT and CoEdit-large models and decode the outputs. Concerning the finetuning stage, we employ AdamW optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0.01$. The learning rate is 2×10^{-5} and the batch size is 32 for each step. We also warm-start the learning for 100 steps. To decode the output of these finetuned models, we use the Beam search algorithm. We set the beam size to 4 and the maximum depth to 64. This search ends immediately when all branches reach the end-of-sentence ($< EOS >$) token. Besides, we use the n-gram approach from Paulus et al. [19] to penalize the repetition of phrases in the beams. We set this parameter to 3. We also promote long sequence outputs from the beam search by setting the exponential penalty to 2.0. All experiments are using the random seed of 42.

With the pre-trained CoEdit [4] model, we use the large version with its default configurations and decoding strategy to produce the output.

6 Evaluation and Results

In this section, we provide detailed explanations of the evaluation metrics used to assess the performance of our models.

6.1 Metrics description

6.1.1 Formality

Due to the subjective nature of formality, we sought a method that could provide a more objective evaluation. To achieve this, we leveraged data from another paper[20], consisting of sentences evaluated by five different individuals for formality.

Data Transformation Steps Here are the steps we used to transform the data to be made usable for formality evaluation:

1. **N-gram Splitting:** Each sentence from the data was split into n-grams with the following sizes: 1,2,3,4,5.
2. **Formality Score Assignment:** Every resulting n-gram had the mean formality score of its original sentence summed.
3. **Summation and Counting:** Alongside the sum of formality scores of n-grams, we maintained a count of each n-gram to facilitate the calculation of mean scores.
4. **Mean Score Calculation:** Combined formality scores for all n-grams were computed, and the mean score for each n-gram was determined and incorporated into the resulting table (termed as "usable n-grams").

Evaluation Steps Through Utilization of Usable N-grams Here are the steps we used to transform the data to be made usable for formality evaluation:

1. **Input Sentence Splitting:** Given an input sentence, it underwent n-gram splitting to facilitate formality score determination.
2. **Formality Score Lookup:** Each n-gram of the input sentence was cross-referenced with the usable n-grams to ascertain its formality score.
3. **Score Aggregation:** The formality scores of individual n-grams were summed and divided by the number of n-grams utilized.
4. **Handling Missing N-grams:** In cases where certain n-grams were absent from the usable n-grams, the total score remained unaffected, and the total number of n-grams for calculating the mean formality score remained unchanged. Essentially, missing n-grams were designed not to influence the overall formality score.

Through this process, we attempted to enhance the objectivity and reliability of our formality evaluations, ensuring a more robust assessment of our model's performance in formality transformation tasks.

6.1.2 BLEU (Bilingual Evaluation Understudy)

For evaluating translation quality, we employed the BLEU metric, a widely recognized measure in the field. Here's how we utilized it:

1. **Reference Comparison:** BLEU evaluates the quality of machine-generated translations by comparing them against one or more reference translations.
2. **N-gram Precision:** Precision is computed by analyzing the presence of n-grams in the generated text compared to reference translations. This process helps quantify the fidelity of translations.
3. **Geometric Mean:** BLEU calculates the geometric mean of precision scores across different n-gram sizes, penalizing shorter translations to ensure a fair assessment.

6.1.3 TER (Translation Edit Rate)

To delve into the accuracy of machine translations, we turned to the TER metric, which measures the extent of edits needed to align machine-generated translations with reference translations:

1. **Edit Distance Calculation:** TER quantifies the edit distance between machine-generated translations and reference translations, encompassing insertions, deletions, and substitutions.
2. **Translation Fidelity:** Lower TER scores signify higher translation quality, indicating fewer edits required to synchronize the two texts and thus better translation fidelity.

6.1.4 TERp (Translation Edit Rate Plus)[21]

As an extension of TER, TERp offers a more comprehensive evaluation by considering phrase-level edits alongside word-level edits:

		BLEU-1	BLEU-2	BLEU-3	BLEU-4	SacreBLEU	TER	TERp	PINC	Formality
BERT2BERT (finetuned)	val	0.00	0.00	0.00	0.00	0.00	15.1907	0.00	1.00	-0.6783
	test	0.00	0.00	0.00	0.00	0.00	14.6883	0.00	1.00	-0.6784
CoEdit-large (pretrained)	val	**82.74**	**74.09**	**66.66**	**59.85**	**59.91**	**0.3339**	**0.0845**	**0.5747**	-0.0565
	test	**82.01**	**73.63**	**66.38**	**59.72**	**59.77**	**0.3300**	**0.0908**	**0.5718**	-0.0517
CoEdit-large (finetuned)	val	2.62	0.95	0.11	0.00	0.00	4.6210	0.0082	0.9984	**0.0415**
	test	2.60	0.91	0.10	0.00	0.00	4.4964	0.0085	0.9993	**0.0306**

Table 4: Performance of models on the raw GYAFC [5] dataset. The best metrics are **bolded** on the validation (**val**) and the **test** set.

1. **Incorporating Phrase-level Modifications:** TERp extends the edit distance calculation to include phrase-level alterations, providing a holistic assessment of translation fidelity.
2. **Comprehensive Evaluation:** Similar to TER, lower TERp scores indicate superior translation quality, reflecting a closer alignment between machine-generated and reference translations.

6.1.5 PINC (Paraphrase In N-gram Changes)

In order to gauge the level of paraphrasing in machine-generated text, we turned to the PINC metric, which quantifies the degree of paraphrasing:

1. **Assessing Paraphrasing:** PINC measures the proportion of n-grams in machine-generated text that deviate from those in reference text, providing insights into the extent of paraphrasing.
2. **Paraphrasing Accuracy:** Lower PINC scores indicate a greater resemblance between machine-generated and reference texts, signifying a higher level of fidelity in preserving original content.

Through the utilization of these metrics, we aimed to obtain a comprehensive understanding of the performance of our models across various dimensions, including translation quality, paraphrasing accuracy, and overall fidelity to reference texts.

6.2 Quantitative Results

Our validation results of the given models on the raw GYAFC dataset are shown in Table 4 while those on the preprocessed dataset are shown in Table 5. It can be seen from the figures that the performance of the finetuned BERT2BERT models is accidentally dreadful, but the Formality criterion seems to be better on the preprocessed data. In contrast, CoEdit models perform well on both sets. Especially, preprocessing the dataset slightly improves its performance. However, finetuning this model deteriorates its result in most of the metrics, except the Formality.

6.3 Qualitative Results

After the evaluation of all of the models developed by us, we found that the BiRNN model with LSTM showed better performance than the General RNN model. Also we found that the finetuned BERT2BERT model exhibited poor performance than the CoEdit model. These findings led us to the conclusion that there is a complex interplay between the model architecture, dataset pre-

		BLEU-1	BLEU-2	BLEU-3	BLEU-4	SacreBLEU	TER	TERp	PINC	Formality
BERT2BERT (finetuned)	val	0.00	0.00	0.00	0.00	0.00	4.8224	0.00	1.00	0.1333
	test	0.00	0.00	0.00	0.00	0.00	4.6628	0.00	1.00	0.1333
CoEdit-large (pretrained)	val	**83.09**	**74.45**	**66.97**	**60.08**	**60.14**	**0.3316**	**0.0837**	**0.5776**	-0.0495
	test	**82.12**	**73.79**	**66.56**	**59.93**	**59.97**	**0.3289**	**0.0900**	**0.5691**	-0.0523
CoEdit-large (finetuned)	val	2.39	0.74	0.08	0.00	0.00	4.7893	0.0035	0.9998	**0.1592**
	test	2.33	0.67	0.00	0.00	0.00	4.6043	0.0042	0.9985	**0.1616**

Table 5: Performance of models on the preprocessed GYAFC [5] dataset. The best metrics are **bolded** on the validation (**val**) and the **test** set.

processing, and finetuning strategies used by us in achieving the optimal performance in Formality Style Transfer.

7 Analysis and Discussions

After conducting experiments, we have several observations and conjectures about the unexpected results.

- The pre-trained CoEdit model produces promising results. This result manifests the well-qualified datasets used to train this model on several downstream tasks.
- Finetuned BERT2BERT and CoEdit models produce unsatisfactory results. This can stem from several reasons:
  - The provided reference does not ensure its formality level, which can down-perform when evaluating.
  - The number of epochs is insufficient to make the BERT2BERT model converge.
  - Finetuning the CoEdit model causes overfitting. This model was already trained on the distilled version of the GYAFC dataset.
  - The configurations for decoding the output are unsuitable for this task. These settings are inspired by Von Platten's work [3], which was conducted on the summarization task.

8 Conclusion

In conclusion, our exploration of various models for formality style transformation has provided valuable insights into their performance and suitability for the task.

Among the models examined, the BiRNN with LSTM demonstrated relatively superior performance in terms of lower loss and improved prediction accuracy for the same example sentence compared to the Genral RNN model developed. This relative advantage underscores the effectiveness of recurrent neural network architectures, particularly those incorporating long short-term memory (LSTM) cells, in capturing and preserving the nuances of formality style transfer.

While further research and optimization are warranted, these findings suggest promising avenues for advancing formality style transformation techniques and enhancing their practical applicability in real-world settings.

[3] https://huggingface.co/blog/how-to-generate

9 Future Work

In the course of our work, we have identified certain areas that merit further investigation and improvement.

- **Improving Formality Scores** One crucial aspect is the need for enhanced formality scores in the context of formality style transfer. The implemented metrics, while valuable, may not capture the intricacies and nuances of formality in a comprehensive manner. The development of more sophisticated formality scores is essential to accurately gauge the formality levels of text. Future work should focus on the exploration of advanced techniques to enhance the precision and reliability of measuring formality scores.

- **Challenges in Measuring Formality**

  Measuring formality itself poses a complex challenge. The multidimensional nature of formality requires a holistic approach that considers various linguistic aspects and contextual cues. As a result, devising a universally applicable and nuanced metric for formality is a non-trivial task. Future research endeavors should delve into the development of comprehensive models that can effectively capture the dynamic and context-dependent nature of formality in diverse textual content.

- **Preparation for Formality Style Transfer**

  Before embarking on formality style transfer applications, it is imperative to ensure the availability of robust metrics. Reliable evaluation metrics are foundational to the success of formality style transfer systems, guiding their development, and enabling meaningful comparisons between different approaches.

- **Model Improvement:** Exploring better model architectures could enhance our formality style transformation models, making them more effective in capturing formality nuances.

- **Data Diversity:** Increasing the diversity of our training data and exploring data augmentation techniques could help address biases and improve model generalization.

- **Fine-Tuning Optimization:** Optimizing our fine-tuning strategies can help improve the performance and robustness of our pre-trained models.

- **Evaluation Metrics:** Expanding our evaluation metrics beyond formality could provide a more comprehensive assessment of our models' performance.

- **Domain-Specific Adaptation:** Investigating how our models can be adapted to specific domains, like legal or medical discourse, could improve their practical usefulness.

- **User Feedback:** Gathering feedback from users and domain experts can help us understand how our models perform in real-world scenarios and guide further improvements.

By focusing on these areas, we aim to enhance the effectiveness and practical applicability of our formality style transformation models.

Bibliography

[1] Filippo Maria Bianchi, Enrico Maiorino, Michael Kampffmeyer, Antonello Rizzi, and Robert Jenssen. *Recurrent Neural Network Architectures*, pages 23–29. 11 2017.

[2] Many to many RNN - Pytorch Implementation. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. Last Accessed: 2024-02-08.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. CoEdIT: Text editing by task-specific instruction tuning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore, December 2023. Association for Computational Linguistics.

[5] Sudha Rao and Joel Tetreault. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[6] Ruochen Xu, Tao Ge, and Furu Wei. Formality style transfer with hybrid textual annotations, 2019.

[7] Zhengyuan Liu and Nancy F. Chen. Learning from bootstrapping and stepwise reinforcement reward: A semi-supervised framework for text style transfer, 2022.

[8] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[9] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994.

[10] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[11] Ondrej 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 Ale s Tamchyna. 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, June 2014. Association for Computational Linguistics.

[12] RNN - Pytorch Implementation. https://pytorch.org/docs/stable/generated/torch.nn.RNN.html. Last Accessed: 2024-02-08.

[13] Abigail See, Peter J. Liu, and Christopher D. Manning. 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, July 2017. Association for Computational Linguistics.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[15] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

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

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[19] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[20] Ellie Pavlick and Joel Tetreault. An empirical analysis of formality in online communication, 3 2016. Submission batch: 10/2015; Revision batch: 12/2015; Published 3/2016.

 Release data:"http://www.seas.upenn.edu/∼nlp/resources/formality-corpus.tgz".

[21] Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TERp system description. Laboratory for Computational Linguistics and Information Processing, Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD 20742, USA; BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA. {snover, nmadnani, bonnie}@umiacs.umd.edu; schwartz@bbn.com.