FormalStyler: GPT based Model for Formal Style Transfer based on Formality and Meaning Preservation

Mariano de Rivero, Cristhiam Tirado and Willy Ugarte[a]

Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Peru

Keywords: Natural Language Processing, Style Transfer, GPT-2, Formalization, Meaning Preservation, Transformer.

Abstract: Style transfer is a natural language processing generation task, it consists of substituting one given writing style for another one. In this work, we seek to perform informal-to-formal style transfers in the English language. This process is shown in our web interface where the user input a informal message by text or voice. This project's target audience are students and professionals in the need to improve the quality of their work by formalizing their texts. A style transfer is considered successful when the original semantic meaning of the message is preserved after the independent style has been replaced. This task is hindered by the scarcity of training and evaluation datasets alongside the lack of metrics. To accomplish this task we opted to utilize OpenAI's GPT-2 Transformer-based pre-trained model. To adapt the GPT-2 to our research, we fine-tuned the model with a parallel corpus containing informal text entries paired with the equivalent formal ones. We evaluate the fine-tuned model results with two specific metrics, formality and meaning preservation. To further fine-tune the model we integrate a human-based feedback system where the user selects the best formal sentence out of the ones generated by the model. The resulting evaluations of our solution exhibit similar to improved scores in formality and meaning preservation to state-of-the-art approaches.

1 INTRODUCTION

Communication is the act of exchanging ideas, thoughts, knowledge, and information between two or more beings. Although communication can be achieved, it may not be in an effective manner. Effective communication is to fulfill the process of communication in the most ideal manner possible by presenting the best style or tone of the content[1]. For instance, conveying ideas in a text with a formal style will raise its quality and it will be perceived more effectively by readers. With the advancements in text-communications over the years and the demand to effectively use time, users have developed tendencies to abbreviate or even omit segments of formal language such as punctuation, capitalization, vocabulary mistakes (e.g., vowel omission), and the use of contractions (Rao and Tetreault, 2018). All the aforementioned are a crucial part of textual formality. The disuse of these conventions highly diminish the quality and formality of a written message.

Our goal is to offer a style transfer model that allows students and professionals, through a web interface, to formalize their written texts. A text with formal style is superior in quality and essential within work and academic environments. We also seek to preserve as much as possible the original message of these texts after the style transfer.

Style transfer is an interesting natural language processing task due to its complexity. The first impediment that appears in style transfer research is the scarcity of resources such as benchmarks, metrics for automatic evaluation, and datasets for training and evaluation. This datasets, in most research to date, are required to be parallel corpus. The most important thing about a style transfer result in a given text is that it preserves the original content that the user desires to transmit, regardless of what style the message had prior to the transfer.

The task of style transfer has recently seen an increase in the field of natural language processing (NLP) and has incurred several improvements in recent years. However, the lack of training resources prevents the development of better solutions. The focus of our project is informal-to-formal text style transfer in the English language. It will be conducted in a web interface that will make the transfers through a trained model.

[a] https://orcid.org/0000-0002-7510-618X

[1] https://bit.ly/3vIOzjn

de Rivero, M., Tirado, C. and Ugarte, W.
FormalStyler: GPT based Model for Formal Style Transfer based on Formality and Meaning Preservation.
DOI: 10.5220/0010674300003064
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 1: KDIR*, pages 48-56
ISBN: 978-989-758-533-3; ISSN: 2184-3228

The core of our research is the GPT-2 pre-trained model which was developed by OpenAI. GPT-2 is a Transformer, which is a deep learning model that uses attention mechanisms (Radford et al., 2019). In this research, we use GPT-2's ability to predict the next word in a text string. To align this task with our desired goal we fine-tune the GPT-2 pre-trained model. This fine-tuning directs its text generation capabilities into a formal style. The formal style is based on the parallel corpus Grammarly's Yahoo Answers Formality Corpus (GYAFC) that contains informal-formal equivalent text pairs (Rao and Tetreault, 2018). We apply a pre-processing step to this corpus which eliminates text pairs which are less than 5 or greater than 25 words.

All the texts generated will be evaluated by two automatic metrics, formality and content preservation. The generated texts are based on user inputs from our web interface. The fine-tuned model uses the inputs to generate multiple possible sentences. The sentences with the highest scores in the formality and meaning preservation metrics are displayed to the user in the web interface. Finally the user selects the best option.

Our contributions are as follows:

- A style transfer model which formalizes informal texts from the English language.
- A functional web interface in which students and professionals will be able to formalize their texts.
- A proposal for a user-feedback constructed dataset.

Our work is structured as: in Section 2 presents the related works. Then, Section 3 presents the background of the project. After, Section 4 presents the main contribution. Finally, Section 5 presents the experimentation and Section 6 presents our conclusions.

2 RELATED WORKS

2.1 Text Generation and Attention Mechanism

Text Generation is the process of building a sentence in natural language for communication with specific purposes. It has been proposed for dialog generation (Serban et al., 2017) and answering. Some approaches include the use of GAN's as in UGAN (Yu et al., 2020a), recurrent neural networks with sequence-to-sequence frameworks for natural language generation as in Multi resolution Recurrent Neural Networks (Serban et al., 2017). Some of the mentioned RNN's include the use of Long-Short-Term-Memory (LSTM) modules that help with feedback connections. This are efficient methods in language modeling tasks such as text generation.

We proposed to use a Transformer-like architecture, that like RNN's are designed to handle sequential input data, the difference being that Transformers adopt Attention mechanism (Vaswani et al., 2017). We use the Transformer-like architecture GPT-2 (Radford et al., 2019) in addition with a formality and content preservation modules in the FormalStyler architecture. We use GPT-2's inherent attention mechanism to assign weights to the words in a input sentence, some words will have higher values which implies that those words are key components in the structure of the input sentence and the content they are ought to deliver. The formal text that we ought to generate in relation to the informal input text will contain the original content of the message in concordance with the key words.

2.2 Text Comprehension

A text have different characteristics that are hard to quantify, not only grammatical rules like orthography, syntax and semantics but also the meaning, intention, style and others (Rao and Tetreault, 2018). Some models have been built with the sole objective of finding if a sentence is grammatically correct (Heilman et al., 2014).

It has to be noted that a text is not only a concatenation of words, but they are also connected making reading comprehension one of the biggest challenges, a model has to process long-term context and remember relevant information (Hoang et al., 2018; Yang et al., 2021; Xu and Cai, 2019). Different sentences sometimes have the same meaning, even if they use completely different words, this is directly related to the interpretation of the sentence, in order to tackle this problem there have been several attempts, some used monolingual parallel corpus (Chen et al., 2019; Pavlick and Nenkova, 2015), extending existent capabilities (Joshi et al., 2018; Srivastava and Jojic, 2018), and others are based on attention mechanisms (Cer et al., 2018). We use the fine-tuned pre-trained embedding given by GPT-2 as input encoding method (Radford et al., 2019).

2.3 Fine Tuning and Evaluation Mechanisms

To fine-tune the pre-trained GPT-2 model from OpenAI we utilize a 110K informal/formal sentence pair dataset from Dear sir or Madam (Rao and Tetreault,

2018) which is used to orientate the general text generation capabilities from the GPT-2 model to a formal text generation model that uses user input informal sentences to begin the formalization task.

Evaluation is a crucial aspect in research specially when analyzing the results of task that are still in a relatively new state such as style transferring. Along the way of the research we utilize the content preservation metric that is proposed in Dear sir or Madam (Rao and Tetreault, 2018) which evaluates the generated formal text from our fine-tuned GPT-2 model, this automatic evaluation scores the generated sentences between 0 and 1, where 0 represents null content preservation and 1 perfect content preservation. Afterwards we decided to use the Universal Sentence Encoder (USE) (Cer et al., 2018) which has the characteristic of representing the meaning of the input sentence on a 512-vector. We take advantage of this characteristic to compare the meaning of the sentences taking the inner product of two 512-vectorial representations and get an accurate content-preservation metric.

We propose a new model method to calculate the formality score based on Dear sir or Madam (Rao and Tetreault, 2018), in this paper the formality is calculated in two ways: using human criteria, and a model (PT16) that has been retrained to increase accuracy.

With the purpose of obtaining similar results as Dear sir or Madams (Rao and Tetreault, 2018) formality metric we developed our own model. This was achieved by using the USE (Cer et al., 2018) and applying transfer learning. The training was done with the same dataset they used and we categorized the formality of the examples based on the conditions proposed. We use the Universal Sentence Encoder (USE) (Cer et al., 2018) as the first layer of this model with three fully connected layers and a single node output with *tanh* activation. This gives us a formality score on the range [-1,1] on a similar way that the previously mentioned paper.

2.4 Style Transfer

There have been several methods of style transfer proposed in recent years. Most of them using some kind of Encoder - Decoder stack (Tian et al., 2018; Prabhumoye et al., 2018). The Attention mechanism is also widely used because of its high efficacy, speed and resource consumption (Gong et al., 2019; Luo et al., 2019a). Some notables approaches have tried to overcome the difficulty of lack of parallel corpus with non-parallel models (Li et al., 2019; Yu et al., 2020b; John et al., 2019).

We propose the use of a fine-tuned pre-trained text generator known as GPT-2 with parallel corpus, that has achieved state of the art results on several specific domain language problems (Radford et al., 2019). Style transfer is difficult due to the high level of complexity of the information that has to be mapped, we parted from the pre-trained model and used it as baseline for our approach.

3 BACKGROUND

In this section we present and explain the main concepts that are used as the foundation of our work. This project aims to train a deep learning model that will be able to change the writing informal style of a user English language text into a formal one. The present work is based on the GPT-2 architecture, which in turn is based on a transformer model. These models make use of an attention mechanism developed specifically to work with sequences and generative models.

3.1 Attention

Attention is a measure that is used to distinguish certain key component words inside a text, giving them a weigh depending on the importance that they possess in relation with the context. Scaled Dot-Product Attention is defined by this formula (Vaswani et al., 2017).

$$Attention(Q,K,V) = softmax(\frac{QK^2}{\sqrt{d_k}}) \tag{1}$$

The input consists of queries and keys of dimension d_k, and values of dimension d_v, where Q, K and V represent matrices, being a the set of queries, set of keys and set of values, respectively. It is a variation of the dot-product algorithm adding the factor $\frac{1}{\sqrt{d_k}}$.

This means that each of the values(V) are multiplied by a weight that determines how each word of the sequence(Q) is affected by the other in sequence(K).

3.2 Transformer

Transformer is an architecture introduced in the paper ‘Attention Is All You Need’ (Vaswani et al., 2017), it makes use of the attention mechanism inside a stack of Encoders / Decoders as seen on Fig. 1. The advantage of this architecture is that it is faster and more efficient because it does not use any Recurrent Neural Network, the information is always feed forward. It compares itself with the previous input to create a chain of inference.

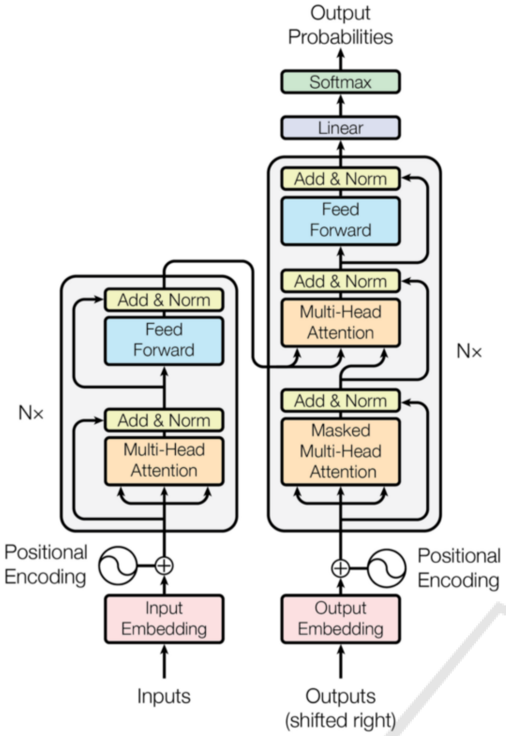

Figure 1: Transformer architecture. From 'Attention Is All You Need' by Vaswani et al.

3.3 Generative Model

A model can be called generative if it generates the next element of a given series, it achieves this by capturing the probability $P(X)$ of the next element, it can be defined as:

$$v_n = max(P(K|V)) \tag{2}$$

Where v_n is the word with the highest probability P on a set of words W given an input V. A variation of this model is choosing one of the top_k words with highest probability. This increases the variability of the model.

4 STYLE TRANSFER MODEL

To begin with, there are four models sizes built for GPT-2 by OpenAI. The Following ones differ in the quantity of parameters in the neural network which are 117M, 345M, 774M and 1.5B. The architecture of the neural network is not compromised regardless of the model size that is used. All of the four sizes models have already been trained to generate text and have state-of-the-art performance in text generation related tasks. Our first and most important objective was to fine tune a GPT-2 to specialize the type of text that it generates, in this case the generation of a formal equivalent content text from a informal one.

4.1 Picking the Model

The first step in our project road-map was to find the biggest model that we could run with the available computational resources at hand, given the limitations of hardware and computer power of our personal machines. ¡. The Pro version, being a paid service, have the following characteristics:

- Up to 24 hour of continuous use per day.
- Up to 25 GB of RAM on request.
- 120 GB of temporary storage per session.
- Priority on Graphic Card allocations.
- TPU allocation on request.

Colab offers two types of runtimes, GPU and TPU, even if TPU is more powerful for heavily load tasks and faster than GPU in this scenarios, GPT-2 repositories only work with GPU based notebooks[2].

In Table 1 we summarize the memory requirements for each of the model sizes for GPT-2. With the computational power available via Google Colab Pro we can run each of the four models. The hurdle that comes at hand is that we are unable to fine tune the biggest model, which requires even more resources than what the Pro version of Colab can offer. We decided to choose the 774M parameter model which can be loaded, executed and fine-tuned on the currently available hardware.

Table 1: Disk and memory space required by each GPT-2 model. $*$ sizes were extracted from direct experimentation $**$ sizes were extracted from (Rajbhandari et al., 2020).

Model	Size	RAM	RAM
		(Execution)	(Fine-tuning)
137M	498 MB*	1.12 GB*	6.64 GB*
345M	1.42 GB*	2.24 GB*	11.42 GB*
774M	3.10 GB*	5.35 GB*	21.08 GB*
1.5B	6.23 GB*	10.44 GB*	60.00 GB**

4.2 Fine Tuning

Once the model has been chosen from the four available, we need to ”teach” it to recognize informal

[2] https://research.google.com/colaboratory/faq.html#gpu-availability

styled sentences and then to formalize them. To accomplish this task, we created a dataset based on the one provided by the paper (Rao and Tetreault, 2018) with the following structure:

```
   <|startoftext|>[Informal]informal
sentence[Formal]formal sentence.
<|endoftext|>
```

We use the flags `<|startoftext|>` and `<|endoftext|>` to inform the model that a sentence has begun or ended respectively. The flags `[Informal]` and `[Formal]` are used to describe the specific sentence style and introduce an ordering.

The model can be loaded, trained, and fine-tuned using either TensorFlow or PyTorch, the required code for this process is complex and extensive so we opted to utilize a wrapper called GPT2-simple which in turn is a wrapper of hugginface GPT2 model that is focused on training GPT-2 like models. The process of training takes around 6 hours to complete and generates a model with a storage space of 3.1GB.

4.3 Model Architecture

Besides doing the style transfer, we also needed to know scores for formality and preservation of content of the results given, we achieve this adding the modules of Content Preservation and Formality as shown in the Fig 2.

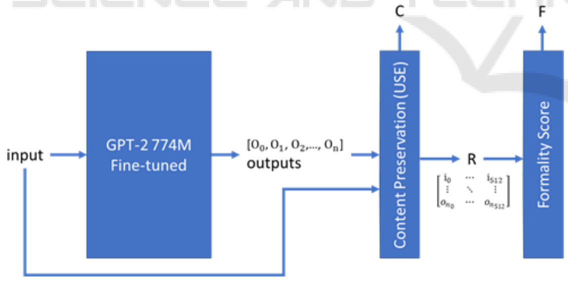
C
F
input
GPT-2 774M
Fine-tuned
$[o_0, o_1, o_2, ..., o_n]$
outputs
Content Preservation (USE)
R
$\begin{bmatrix} i_0 & \cdots & i_{512} \\ \vdots & \ddots & \vdots \\ o_{n_0} & \cdots & o_{n_{512}} \end{bmatrix}$
Formality Score

Figure 2: FormalStyler model architecture.

Content Preservation: To analyze it we must compare the content of the outputs with the input. We use the version 5 of the Universal Sentence Encoder (USE), trained by Google (Cer et al., 2018) to encode the inputs and outputs as 512-vectors with values on the range [-1, 1]. Then we use the inner product of the vectors to find the similarity between those two defined as C in Eq. 3. C_i is the preservation score for the $i-th$ output where R_x is the vectorial representation of x.

$$C_i = R_{input} \cdot R_{output_i} \tag{3}$$

Formality: In order to recognize the formality level on an automatic way, we used the representations of the training dataset given by the USE to train a three layer fully connected model with a single output node that returns the formality score (F). The activation `tanh` ensures that the output is in the range [-1,1] providing two metrics on one score. The architecture of this module is described on Fig. 3:

This score is a useful way of representing the formality of a sentence and helping to quickly understand the classification of the word. This score has the following rules:

- If the score is >0 then the sentence can be considered formal; if its <0, is considered informal.
- The closest the absolute value of the score is to 1, the higher the level of formality or informality.
- We can assume a sentence with value zero is neutral on formality.

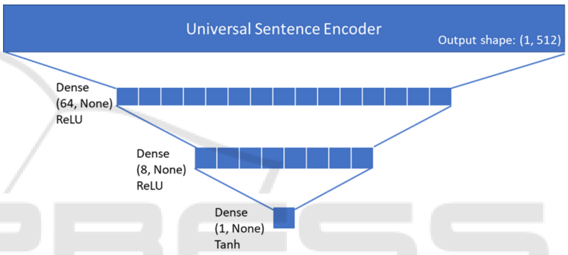

Figure 3: Formality module architecture.

4.4 Deployment

The model was deployed and working on a single-user environment where all the computational power was assigned to an individual. In order to understand the limitations and potential of the model, we decided to deploy it as an API to test concurrency and increase the quantity of queries.

- *Queue:* We implemented a queue where every query was assigned to the model in order. The downside of that approach is that every query takes 30 seconds on average, so with ten concurrent tests, the last one would have to wait more than five minutes for the transfer.
- *Parallel Execution:* Another method was trying to deploy models on-demand, every time a query was received, a new model was deployed, and after the calculation it was discarded.

5 EXPERIMENTATION

In this section we present our experimental study to show the results of our informal-to-formal style transfer model which attains similar results as the state of the art.

5.1 Experimental Protocol

For the development of our informal-to-formal style transfer model we used the following resources:

1. **Software:**
   - Python 3.7
   - Tensorflow 1.15.2
   - Google Colab Pro
   - For the fine-tuned style transfer model:
     - GPT-2 pre-trained model in 4 sizes: Mini, Small, Medium and Big (Hyperparameter amount difference)
     - Grammarly's Yahoo Answer Formality Corpus (GYAFC)
2. **Hardware:** To solve this limitation we opted to use Google Colab Pro service which offered the following resources:
   - Storage: SSD 125GB
   - RAM: 24GB
   - GPU: Nvidia® Tesla V100-SXM2 16 GB
   - CPU: Intel® Xeon® CPU @ 2.20GHz
3. **Dataset:** We used the GYAFC Dataset (Rao and Tetreault, 2018) this corpus has the following characteristics:
   - **Sentences:** 314,314
   - **Pairs:** 157,157
   - **Main subjects:** Family, relationships and entertainment.

Our code is publicly available at: github.com/TBinc/FormalStyler

5.2 Results

The model and the metrics were evaluated in order to determine their performance and properties. We focused on three factors: style transfer, meaning preservation and formality.

5.2.1 Style Transfer

The main objective of this model is to generate a formal sentence based on an informal one, without changing the meaning. We achieved this by fine-tuning the GPT-2 774M model (Radford et al., 2019). This model has the following characteristics:

- **Embedding size:** 1,280
- **Vocabulary size:** 50,257
- **Context size:** 1,024
- **Layers:** 12

The fine-tuning of the model took 6 hours with 3,000 steps using the GYAFC Dataset.

Due to the parallelized nature of GPT-2, it is easy to generate multiple outputs given a single input, but if we want to use multiple inputs, we have to wait for the previous execution to finish.

Table 2: Execution time for style transfers in seconds.

	Number of transfers per input				
Inputs	1	2	4	6	8
1	18.55	19.91	21.39	23.44	25.91
2	37.14	39.49	42.43	46.63	51.91
3	55.69	59.40	63.83	70.07	77.82

As shown in Table 2, the time it takes the model to transfers the style has a linear behavior. We model this behavior on equation 4, where t is the time in seconds for the execution, i is the number of inputs and o represents the number of transfers generated for an input i.

$$t(i,o) = \sum_{j=1}^{i}(1.0137 * o_j + 17.595) \tag{4}$$

5.2.2 Formality

As stated previously we developed our own model to calculate the formality of the sentence, we trained this model with the same dataset used for the fine-tuning.

We used this model to evaluate the the formality of the outputs of the style transfer and its characteristics. We used 2518 manually transferred sentences to evaluate the performance of the model. Table 3 contain some statistical information regarding the results on the formality metrics of our model.

In Figure 4 we can see the histogram of the scores of the result, most of the scores are above zero and the biggest group has a score higher than 0.7.

5.2.3 Meaning Preservation

The meaning preservation was calculated as the inner product of the vectors generated by the Universal Sentence Encoder (Cer et al., 2018). Similar to the formality of the outputs of the style transfer and its characteristics, we present this results on Figure 5 in the form of an histogram and the numeric data on Table 3.

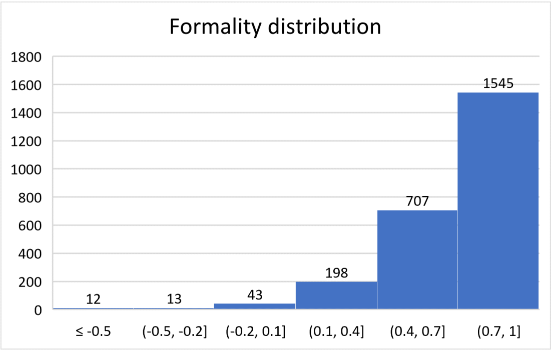

Figure 4: Formality score distribution.

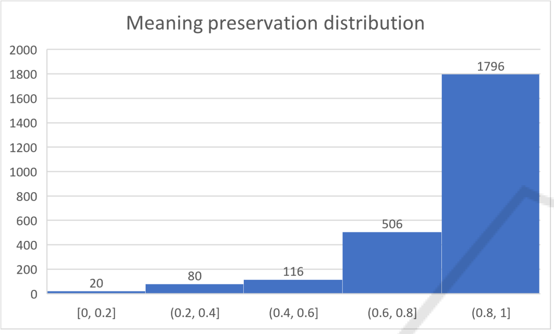

Figure 5: Meaning preservation score distribution.

Table 3: Summary of Formality and Meaning Preservation metric results.

	Score	
	Formality	Meaning Preservation
max	1.0000	1.0000
mim	-0.804	0.1316
mean	0.7312	0.8468
std	0.2394	0.1766

5.2.4 Model Size Impact

We fine-tuned the 774M GPT-2 model to develop the present model, we also trained the models 137M and 345M. In Table 4 we present the comparison between models.

It can be noted that the size of the model has a direct impact on the metrics. The difference between the 774M y 137M models are 14.96% in the Formality score and 7.94% in the Meaning Preservation score. This means that the size of the model has a bigger impact on the formality than in the meaning preservation.

Table 4: Formality and meaning preservation metric results by model size.

	Score	
Model Size	Formality	Meaning Preservation
137M	0.6218	0.7796
345M	0.6677	0.8051
774M	0.7312	0.8468
1.5B	0.8123	0.9045

With those results we can project the possible metrics for the 1.5B model, we assumed that the growth factor has a linear behavior, this is because the sizes of the models are almost doubled every time.

5.2.5 Benchmark

We compared our results with three state of the art approaches. THe metric of meaning preservation was multiplied by six to make it comparable and the Overall score was calculated by equation 5.

$$O = 2 * \frac{F * \frac{M}{6}}{F + \frac{M}{6}} \tag{5}$$

Where O is the Overall score, F corresponds to formality and M is meaning preservation.

Table 5: Benchmark between different models published.

Model	**Form.**	**Mean. Pres.**	**Overall**
Our approach	0.73	**5.08**	0.78
(Gong et al., 2019)	**0.83**	4.96	**0.83**
(Wu et al., 2019)	0.73	4.56	0.75
(Luo et al., 2019b)	0.61	3.62	0.61

We can note the our approach has a state of the art meaning preservation score but lacks in formality. This can be due to the model size used.

5.3 Discussion

In Figure 4 we can see that 61.36% of the output sentences have a score higher that 0.7, we can consider those sentences as highly formal, and in Figure 5 it can be noted that 71.33% of the transfers got an score of 0.8 or higher, which is to say that the meaning was preserved between the input and the output. Besides the efficiency of the model, this. This is due to the high efficiency of the pre-trained model (Radford et al., 2019) and the successful application of the fine-tuning.

Due to lack of resources we couldn't fine-tune the 1.5B GPT-2 model, this model has performed better than the smaller ones in a variety of scenarios (Radford et al., 2019). The use of this model could potentially increase the formality and meaning preservation of the transfers, the approximate scores would have been a 0.85 for formality and 0.9 for meaning preservation.

The histograms depicted on Figures 4 and 5 present a skewed left distribution, this means that most of the scores on formality and meaning preservation have a very high value in the range.

6 CONCLUSIONS

Thanks to this research we have established that successful informal-to-formal style transfer tasks that presents high scores in formality and meaning preservation can be accomplished by fine-tuning a pre-trained Transformer model like the GPT-2 (Radford et al., 2019), being the GPT-2 original task to handle sequential input data, with a parallel corpus and connecting it with a meaning preservation and formality modules.

The lack of training input data impacted directly in the fine-tuning procedure that we applied to the GPT-2 pre-trained model (Radford et al., 2019), for instance the GYAFC parallel corpus (Rao and Tetreault, 2018) with its 110k informal/formal sentence pairs was enough, by a small margin, to produce the desired results that we obtained. If the parallel corpus used to perform the fine-tuning procedure had been smaller our final results would have been strictly inferior both in its meaning preservation and formality scores.

The use of transformers is recommended for Natural Language Processing, specially in Style Transfer tasks, due to its attention mechanisms which weight the influence of different parts of input data. A different allocations of the fully connected layers could potentially decrease the computational time required for the style transfer, which would consequently diminish the time resources needed. Using the biggest One-Shot model, like the 1.5B pre-trained model of GPT-2 (Radford et al., 2019) or Few-shot learning model, like the GPT-3 (Brown et al., 2020), would potentially outperform in all steps of the process in style transfer tasks and generate better results.

Our approach for Style Transfering might be used for Question Answering for HRI (Burga-Gutierrez et al., 2020) or furthermore using softness for tunning the meaning preservation metrics (Ugarte et al., 2015).

ACKNOWLEDGMENT

We would like to warmly thank Joel Tetreault, co-author of the paper *Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer* for providing us with the parallel corpus dataset *Grammarly's Yahoo Answer Formality Corpus" (GYAFC)* which is based on the L6 corpus of Yahoo Answers.

REFERENCES

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *NeurIPS*.

Burga-Gutierrez, E., Vasquez-Chauca, B., and Ugarte, W. (2020). Comparative analysis of question answering models for HRI tasks with NAO in spanish. In *SIMBig*, pages 3–17. Springer.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder for english. In *EMNLP (Demonstration)*, pages 169–174. Association for Computational Linguistics.

Chen, X., Zhang, M., and Zhu, K. Q. (2019). Aligning sentences between comparable texts of different styles. In *JIST (2)*, volume 1157 of *Communications in Computer and Information Science*, pages 51–64. Springer.

Gong, H., Bhat, S., Wu, L., Xiong, J., and Hwu, W. W. (2019). Reinforcement learning based text style transfer without parallel training corpus. In *NAACL-HLT (1)*, pages 3168–3180. Association for Computational Linguistics.

Heilman, M., Cahill, A., Madnani, N., Lopez, M., Mulholland, M., and Tetreault, J. R. (2014). Predicting grammaticality on an ordinal scale. In *ACL (2)*, pages 174–180. The Association for Computer Linguistics.

Hoang, L., Wiseman, S., and Rush, A. M. (2018). Entity tracking improves cloze-style reading comprehension. In *EMNLP*, pages 1049–1055. Association for Computational Linguistics.

John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. (2019). Disentangled representation learning for non-parallel text style transfer. In *ACL (1)*, pages 424–434. Association for Computational Linguistics.

Joshi, V., Peters, M. E., and Hopkins, M. (2018). Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL (1)*, pages 1190–1199. Association for Computational Linguistics.

Li, D., Zhang, Y., Gan, Z., Cheng, Y., Brockett, C., Dolan, B., and Sun, M. (2019). Domain adaptive text style transfer. In *EMNLP/IJCNLP (1)*, pages 3302–3311. Association for Computational Linguistics.

Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sun, X., and Sui, Z. (2019a). A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*, pages 5116–5122. ijcai.org.

Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sun, X., and Sui, Z. (2019b). A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*, pages 5116–5122. ijcai.org.

Pavlick, E. and Nenkova, A. (2015). Inducing lexical style properties for paraphrase and genre differentiation. In *HLT-NAACL*, pages 218–224. The Association for Computational Linguistics.

Prabhumoye, S., Tsvetkov, Y., Black, A. W., and Salakhutdinov, R. (2018). Style transfer through multilingual and feedback-based back-translation. *CoRR*, abs/1809.06284.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. (2020). Zero: memory optimizations toward training trillion parameter models. In *SC*, page 20. IEEE/ACM.

Rao, S. and Tetreault, J. R. (2018). Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *NAACL-HLT*, pages 129–140. Association for Computational Linguistics.

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. C. (2017). Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, pages 3288–3294. AAAI Press.

Srivastava, S. and Jojic, N. (2018). A spatial model for extracting and visualizing latent discourse structure in text. In *ACL (1)*, pages 2268–2277. Association for Computational Linguistics.

Tian, Y., Hu, Z., and Yu, Z. (2018). Structured content preservation for unsupervised text style transfer. *CoRR*, abs/1810.06526.

Ugarte, W., Boizumault, P., Loudni, S., Crémilleux, B., and Lepailleur, A. (2015). Soft constraints for pattern mining. *J. Intell. Inf. Syst.*, 44(2):193–221.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS*, pages 5998–6008.

Wu, C., Ren, X., Luo, F., and Sun, X. (2019). A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *ACL (1)*, pages 4873–4883. Association for Computational Linguistics.

Xu, J. and Cai, Y. (2019). Incorporating context-relevant knowledge into convolutional neural networks for short text classification. In *AAAI*, pages 10067–10068. AAAI Press.

Yang, J., Zhang, Z., and Zhao, H. (2021). Multi-span style extraction for generative reading comprehension. In *SDU@AAAI*, volume 2831 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Yu, W., Chang, T., Guo, X., Wang, X., Liu, B., and He, Y. (2020a). UGAN: unified generative adversarial networks for multidirectional text style transfer. *IEEE Access*, 8:55170–55180.

Yu, W., Chang, T., Guo, X., Wang, X., Liu, B., and He, Y. (2020b). UGAN: unified generative adversarial networks for multidirectional text style transfer. *IEEE Access*, 8:55170–55180.