Tweet to News Conversion: An Investigation into Unsupervised Controllable Text Generation

Zishan Ahmad*§, Mukuntha N S†§, Asif Ekbal‡ and Pushpak Bhattacharyya¶
Department of Computer Science and Engineering, Indian Institute of Technology Patna
Bihta, Patna
Email: *1821cs18@iitp.ac.in, †mukuntha.cs16@iitp.ac.in, ‡asif@iitp.ac.in, ¶pb@iitp.ac.in

***Abstract*—Text generator systems have become extremely popular with the advent of recent deep learning models such as encoder-decoder. Controlling the information and style of the generated output without supervision is an important and challenging Natural Language Processing (NLP) task. In this paper we define the task of constructing a coherent paragraph from a set of disaster domain tweets, without any parallel data. We tackle the problem by building two systems in pipeline. The first system focuses on unsupervised style transfer to convert the individual tweets into news sentences. The second system stitches together the outputs from the first system to form a coherent news paragraph. We also propose a novel training mechanism, by splitting the sentences into propositions and training the second system to merge the sentences. We create a validation and test set consisting of tweet-sets and their equivalent news paragraphs to perform empirical evaluation. We also perform human evaluations on our model. In a completely unsupervised setting our model was able to achieve a BLEU score of 19.32, while successfully transferring styles and joining tweets to form a meaningful news paragraph.**

***Index Terms*—Deep-learning, Unsupervised, Style-transfer, Text generation**

I. Introduction

Text generation using neural networks has become immensely popular in recent times due to the advent of sequence-to-sequence (seq2seq) [1], [2] models. Long Short Term Memory (LSTM) [3] and modern Transformer [4] based models have been revolutionary for several tasks like machine translation, abstractive text summarization etc. However, the training of such models require a huge amount of data, which is expensive and cumbersome to create. Although there are available large datasets for a few text generation tasks, creation of a dataset for every new task is not always practical or feasible. To get around this issue, unsupervised text generation is increasingly being used, particularly in the area of Machine Translation [5] and Style Transfer [6].

Another problem with text generation is the control of information being generated. Models like GPT-2 [7] produces long paragraphs resembling human-written text, that are fluent to read. However, the input to the model is only the starting few words of the sequence[1]. The generated paragraph-although fluent, does not necessarily represent the information that we may want to convey. Thus guiding the text generator to generate the information that is desired, and in the style that it is desired in, is a crucial task in text generation. Doing so in an unsupervised setting is a non trivial task, that poses the following challenges for the system.

- The system should be able to achieve a good mapping between the input style and the desired style in the output without any supervision.
- The system should not generate extra information or miss some information that is provided at the input.
- The system should learn to stitch together the nuggets of information provided at the input, to produce a coherent paragraph in the desired style. This too should be achieved without any parallel data.

In this paper we define a new task of converting a set of tweets into a news paragraph to explore the above challenges. The task not only poses a great challenge to the field of Machine Learning and Natural Language Processing, but is also an important task in terms of applications. The system can be used to automatically generate news from the live tweet-stream. This can be crucial in emergency situations like disasters, for gaining comprehensive live situational data, in a non-noisy, readable, formal text format. Since we do not have any parallel data for the task, we have to develop unsupervised techniques to solve this problem. We try to tackle the problem by dividing it into two parts, **i) Part 1:** Converting individual tweets to formal news text format, and **ii) Part 2:** Stitching together the converted tweets to form a comprehensive and fluent news paragraph that contains all the information provided in the tweets.

We build two separate systems to tackle Part 1 and Part 2. The entire system is structured in a pipeline where Part 1 would first take the input tweets, and then convert them into news format. The second part (i.e. Part 2) would then take the converted tweets from the first part and output a coherent news paragraph that contains information from all the tweets. For Part 1, we model the task as an unsupervised style transfer task of transferring from tweet-style to news-style. Although unsupervised style transfer has been explored before, it has mostly been carried out to change the sentiment of a text [6], [8]. Li et al. [9] show that sentiment style transfer can be achieved by only changing a few attribute markers (such as 'good' to 'bad'), while leaving the sentences largely unchanged. Thus, it is easier to capture sentiment pattern in text than the pattern that defines informal (tweet)

§equal contribution
[1]https://talktotransformer.com/

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

or formal text (news). Such a task is more complex, requiring injecting articles wherever necessary, replacing informal words with formal words, discarding hashtag words where it is not important, retaining the hashtags wherever important and getting rid of other noise in the text as well, for which rules cannot be written.

The second part (i.e. Part 2) is tackled by a separate system that takes the outputs generated by the first part and joins them together to generate a paragraph. The task to be tackled in Part 2 is again non trivial, as it has to get rid of redundancies in information in different tweets, and keep the unique information. While doing so, it also has to figure out a way of joining sentences wherever required and not joining sentences when not required. In this paper, we develop a system that only learns to join sentences together. We propose a novel unsupervised approach to achieve joining of separate sentences, where we use news sentences that have been broken down into propositions. After obtaining news sentences and their clauses, we train an encoder-decoder based model that learns to stitch these propositions together to form a coherent sentence. We also create and release a validation and test set for the task to quantitatively measure the system's performance.

Our evaluation shows that our method performs well in terms of both automatic and human evaluation metrics.In summary, our current work contributes:

- A new method to convert informal and noisy tweet content into a more formal news-text format.
- A novel training regime to stitch together information nuggets into long coherent sentences.
- An evaluation dataset for the task, consisting of 1265 instances of a set of four disaster-domain tweets and their corresponding news paragraphs.

A. Problem Definition and Motivation

Given a set of four tweets of a particular event from the disaster domain, the task is to form a paragraph in news format that contains all the information provided in the tweets and does not produce extra information. The model should thus form a formal and comprehensible news paragraph from the given noisy input (i.e. tweets). Following is an example of the task:

- **Tweet 1 (Input 1):** breaking: at least 126 killed in taliban attack on pakistani school.
- **Tweet 2 (Input 2):** update: #peshawarattack over, all hostage-takers dead - police.
- **Tweet 3 (Input 3):** dozens of children killed as taliban gunmen storm peshawar school.
- **Tweet 4 (Input 4):** #pakistan school attack over, all six attackers are dead. #peshawarattack #talibanattacksschool
- **News (Output):** Taliban gunmen stormed a school in Pakistan, killing at least 126. Police stated that the Peshawar attack is over as all six hostage-takers are dead.

The above problem is defined to tackle the challenges mentioned in Section I. The difficulty of this task is also exacerbated as tweets are extremely noisy input source.

This task is motivated to address important NLP challenges such as information guided, style controlled unsupervised text generation. Also, joining pieces of information to generate a well formed paragraph has been largely unexplored. The system built to solve this task has useful application as an automated news generator, that takes live tweets as inputs and generates news paragraph. This is useful especially in disaster domain, as it can provide better situational awareness. The users will not have to explore endless noisy tweets but can get one news paragraph that provides live information in a clear news format. This system, thus, helps us take another step in automated journalism.

II. Related Work

The task defined in this paper has not been previously tackled, to our knowledge. However, there are a few tasks that aim to solve a few sub-tasks and adjacent tasks. One of the sub-tasks that has been previously been explored is 'style-transfer in text'. Nahas et al. [10] used LSTM based attentive encoder-decoder network to style transfer between old and modern Turkish text. They had parallel corpus between old and modern Turkish language to train their model. Since it's often not possible to obtain parallel corpora, semi-supervised and unsupervised text style transfer are also very actively being explored in the machine learning community. Shang et al. [11] proposed a semi-supervised method of transfer text style between Chinese ancient poetry style and modern Chinese style. They also tested their experiment on style transfer between formal and informal English text. Unsupervised style transfer is often achieved by using unsupervised neural machine translation techniques [6], such as back-translating data in one style to create synthetic parallel data. Jin et al. [12] introduced iterative matching of synthetic data produced by unsupervised Neural Machine Translation (NMT) technique. They used cosine similarity and word mover distance to keep the best synthetic parallel data. They showed that their method outperforms unsupervised NMT method in sentiment transfer and formality transfer tasks. Yang et al. used [8] language models to discriminate between the sentiment of the generated text by unsupervised NMT method. This way they were able to propagate error at every step of generation and achieve better performance in transferring sentiment of the text. Luo et al. [13] used dual reinforcement learning to better capture style and content and improve the style transfer. They evaluate their model for sentiment and formality transfer tasks.

Another task that may seem similar to our task is text summarization. Traditional text summarization tasks aim to condense long text into a small summary. This summary may even exclude some information. In our task, input information should not be lost at the output, even though redundancy has to be removed. Also the length of the output text may or may not be shorter than the input text. Abstractive summarization tasks require a huge amount of parallel data. Various tasks like headline generation [14], [15], wikipedia summarization [16] and news document summarization [17] have been tackled using supervised abstractive text summarization. Encoder-

decoder neural models are the most popular techniques for abstractive summarization. Some works like the one presented in Liu et al. [16] use discriminator to identify human and machine created text, to improve the performance of their encoder-decoder model.

Unsupervised summarization is usually achieved through extractive summarization techniques. Handcrafted features are often used [18]–[20] to extract sentences. These extracted sentences are assumed to represent the summary of the text. Shen et al. [21] used trained a Conditional Random Field (CRF) classifier to label sentences as part of summary or not part of summary. Although extractive summary, this method still needed labelled data.

The advantage of extractive text summarization is that it can be achieved in unsupervised manner. However, a major drawback with extractive text summarization techniques is that the summaries formed are not coherent and are disjoint. The summary formed is not a fluent paragraph, but a collection of sentences that represent different information. Abstractive summarization forms well formed, comprehensible summaries, however requires a huge amount of parallel data. In our knowledge there has been no prior work that performed unsupervised abstractive text summarization. Although the work in this paper is different from summarization, it still tackles the problem of removing redundancies and joining sentences to form a coherent paragraph.

III. Proposed Methodology

As discussed in Section I, we solve the problem in two parts and build separate system for each part. In part 1, we model the problem as a style transfer task from tweets to news, whereas in part 2, the output from part 1 is taken and stitched together to form a paragraph. Throughout our experiments, we use a transformer model [4] as our encoder-decoder model, and initialize it with the weights of cross-lingual language model (XLM) [22]. The details of each system is discussed in this section.

A. Cross-lingual language model (XLM)

Lample et al. [22] proposed XLM to create a transformer [4] based cross-lingual language model. To achieve this, three different training mechanisms are followed:

1) **Causal Language Modeling (CLM):** CLM is modelled to predict the probability of the next word given the previous context in the sentence $P(w_t|w_1, w_2, ...w_{t-1}, \theta)$.
2) **Masked Language Modeling (MLM):** MLM [23] is achieved by randomly sampling 15% of the input BPE (byte-pair encoded) tokens. These sampled tokens are replaced by: i). [MASK] token 80% of the time, ii). Random token 10% of the time and iii). Are left unchanged 10% of the time.
3) **Translation Language Modeling (TLM):** The TLM is modelled as an extension of MLM. Here, instead of using only monolingual text-streams, parallel sentences of different languages are concatenated. Random masking is then performed in all the sentences. To predict a masked word in one language, the model can either look at the surrounding context of the same language sentence, or it can look into parallel sentences of other language to get hint of the mask word. This is the way how the model learns to find alignment between the different languages.

The XLM model uses only one encoder and one decoder with the language embeddings of several languages, instead of using separate encoders and decoders to create a multilingual system. Thus, this model consumes less memory than multi-encoder-decoder systems. In our experiments, since we do not have any parallel data, we only use MLM. To use this model, we treated tweets and news as two styles, and train separate style embeddings for them, analogous to the language embeddings used in XLM.

B. Proposed Model-First Part

To solve the style transfer task in the first part (i.e. Part 1), two different systems are built. We take two non-parallel datasets of two different styles, $X = \{x_1, x_2 \dots x_m\}$ consisting of tweets and $Y = \{y_1, y_2 \dots y_n\}$ consisting of news sentences. Let l_1 denotes the tweet-style and l_2 denotes the news-style. Our goal is to model the conditional distributions $p(x|y)$ and $p(y|x)$; i.e., to transfer data x of the tweet style l_1 to the news style l_2. The encoder E encodes inputs x and y to give content vectors $z_x = E(x, l_1)$ and $z_y = E(y, l_2)$. The decoder D decodes z_x and z_y to give $\hat{x} = D(z, l_2)$ and $\hat{y} = (z, l_1)$ respectively, where z is any content vector outputted by the encoder. Also, let θ_E and θ_D represent the parameters of the encoder and decoder, respectively.

1) XLM-STY: We use unsupervised neural machine translation (UNMT) steps [24] to achieve style transfer between tweets and news. The steps followed in this model are listed below:

- *De-noising step:* In this step noise in the form of random masking, shuffling and dropping of BPE tokens is added to the encoder input sentence, while the decoder is trained to reconstruct the original de-noised sentence. This is done in order to make the model learn the two style distributions. The reconstruction loss L_{rec} is given in Equation (1). Here, $x \in X$ is a tweet, and $y \in Y$ is a news sentence. $\hat{x}_{dn}$ $D(E(C(x), l1), l1)$ denotes that $\hat{x}_{dn}$ is a reconstruction of the noised version $C(x)$ of the input tweet x, where C is a function that adds random noise to x.

$$L_{rec}(\theta_E, \theta_D) = \mathbb{E}_{x,\hat{x}_{dn} \sim D(E(C(x),l_1),l_1)}(-log(p(\hat{x}_{dn}|x))) + \mathbb{E}_{y,\hat{y}_{dn} \sim D(E(C(y),l_2),l_2)}(-log(p(\hat{y}_{dn}|y))) \quad (1)$$

- *On-the-fly-back-translation:* While the de-noising step helps learn the individual style distributions, it does not help learn the transfer of a sentence from one style to the other. So in this step, we use our current style-transfer model M_{12} to a news sentence x to obtain a synthetic version $y' = M_{12}(x)$ in the target tweet domain. These are then used as parallel data to help

train the encoder and the decoder in the tweet-to-news direction. The same is repeated with a tweet y and a generated synthetic news sentence $x' = M_{21}(y)$ to train the model in the news-to-tweet direction. Here, the style-transfer models from which synthetic outputs are sampled, are given by $M_{12}(x) = D(E(x, l_1), l_2)$ and $M_{21}(y) = E(D(y, l_2), l_1)$. Equation (2) defines the back-translation loss L_{bt}.

$$L_{bt}(\theta_E, \theta_D) = \mathbb{E}_{x,\hat{x}_{bt}\sim D(E(M_{12}(x),l_2),l_1)}(-log(p(\hat{x}_{bt}|x))) + \mathbb{E}_{y,\hat{y}_{bt}\sim D(E(M_{21}(y),l_2),l_2)}(-log(p(\hat{y}_{bt}|y))) \quad (2)$$

2) XLM-STY-DIS + SYN: This model introduces two new modifications to the baseline *XLM-STY* model (c.f. Figure 1):

- *Adversarial training (DIS):* We additionally train a Gated Recurrent Unit (GRU) based discriminator D that takes in the content vectors $z_x = E(x, l_1)$ and $z_y = E(y, l_2)$ produced by the encoder and classify them as tweet or news. This is done in order to obtain a style-invariant representation at the encoder's output, thus aligning the two z distributions. This style invariant content representation is then fed to the discriminator, which decodes the representation into the desired style. The discriminator is trained as a binary classifier that outputs the probability $p(l_1|z)$ that a given latent content vector z comes from a tweet, with $p(l_2|z) = 1 - p(l_1|z)$ being the probability that it comes from a news sentence. The adversarial loss L_D used to train the discriminator weights θ_D is given by the following cross-entropy loss:

$$L_D(\theta_D|\theta_E) = \mathbb{E}_{x\sim X, z_x\sim E(x,l_1)}(-log(p(l_1|z_x))) + \mathbb{E}_{y\sim Y, z_y\sim E(y,l_2)}(-log(1 - p(l_1|z_y))) \quad (3)$$

The encoder is trained with the opposite objective:

$$L_{adv} = -L_D \quad (4)$$

- *Synthetic parallel-data training step (SYN):* To make the model more robust, we add another step to the training process. Apart from *UNMT* and *Adversarial Training* step, the encoder-decoder model is also trained with a synthetic tweet data that we create. Tweets have some unique properties [25], like random spelling mistakes, random hashtags, hashtags that convey important information etc. These properties cannot be induced by our synthetic tweet generator, as the generator is unlikely to produce random spelling mistakes, and also cannot generate random hashtags. We try to mimic these properties by creating a synthetic parallel tweet-data using news sentences and following the steps below:
  - **Step 1:** Paraphrase the English news sentence by using pre-trained translation modules to translate it to a distant language and translating it back to English. This way the paraphrasing is a little noisy, which is helpful for our task.
  - **Step 2:** Introduce random spelling mistakes like flipping random characters, dropping random characters and dropping all the vowels from the word with a probability of 15%.
  - **Step 3:** Randomly hashtag named entities from the sentences with a probability of 15%. Also we have a list of random hashtags used from the tweet corpus. Using this list we randomly inject a hashtag into each tweet with a 15% probability.

  This synthetic data is used to train the same encoder-decoder model in a separate step. The reconstruction loss L_{rec} used for this training is given in Equation 5. Here $H(x)$ is the function that converts the news sentences into synthetic tweets by following the aforementioned steps.

$$L_{syn}(\theta_E, \theta_D) = \mathbb{E}_{y,\hat{y}_{syn}\sim D(E(H(y),l_2),l_1)} - log(p(\hat{y}_{syn}|y)) \quad (5)$$

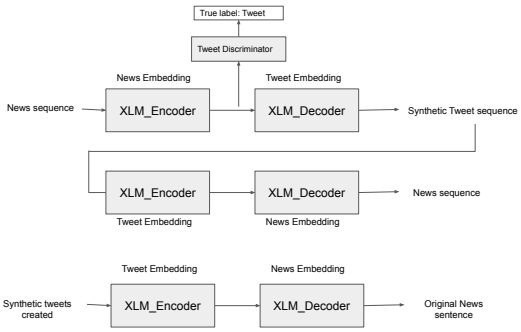

Fig. 1. Block diagram of the 'XLM-STY-DIS + SYN' architecture

C. Part 2

To tackle part 2, we train XLM-MERGE, a model capable of stitching together the information nuggets from the news-style sentences obtained from Part 1. To achieve this, we break the news sentences into propositions using ClauseIE [26]. ClauseIE is a clause-based framework to open information extraction from sentences. It is based on dependency parse trees, and extracts complex propositions from news sentences. Given a news sentence y, ClauseIE extracts a set of 'propositions' $p_1, p_2, \ldots p_n$. Each proposition extracted consists of a subject phrase (for example "a young woman"), a relation phrase ("has been airlifted") and zero, one or more argument phrases ("to hospital"). We join these propositions to form sentences ("a young woman has been airlifted to hospital"). We then train an encoder-decoder architecture to merge these propositions back into their original sentence with the following cross-entropy loss L_m.

$$L_m(\theta_E, \theta_D) = \mathbb{E}_{y,\hat{y}_m\sim D(E(P(y),l_1),l_2)}(-log(p(\hat{y}_m|y))) \quad (6)$$

Here, $P(y) = p_1\|p_2\|\ldots\|p_n$, a paragraph of appended proposition sentences extracted from the news sentence y. $\hat{y}_m$ is a reconstruction of the original sentence from $P(y)$. We initialize both encoder and decoder with the pretrained XLM Masked Language Model. Let θ_E and θ_D be the parameters of the encoder and decoder, respectively, and l_1 and l_2 be the

TABLE I
DATA DISTRIBUTION

Dataset	Number of Instances
News-Tweet-Parallel Validation	265
News-Tweet-Parallel Test	1,000
Tweet-Non-Parallel	45,295
WMT-Non-Parallel	171,400
News-Clause-Pair	126,120

language-embeddings used for propositions and full sentences, respectively.

The following is an example of propositions extracted by ClauseIE:

- **Sentence**: a young woman has been airlifted to hospital after her car veered into trees in the la trobe valley.
- **List of Propositions Extracted**: a young woman has been airlifted to hospital. a young woman has been airlifted after her car veered into trees in the la trobe valley. a young woman has been airlifted into trees in the la trobe valley. a young woman has been airlifted in the la trobe valley. she has car. her car veered after her car veered into trees in the la trobe valley. her car veered into trees in the la trobe valley. her car veered in the la trobe valley

As we can see from the above example, the propositions contain redundant information, and are sometimes noisy. This is very similar to tweets. While a majority of the propositions are useful and help us regenerate the sentence, we also find that a few of them can be incorrect ("a young woman has been airlifted into trees in the la trobe valley."). However, this noise is only present at the source-side and makes the model more robust to noisy data coming from Part 1. Unsupervised machine translation methods often employ back-translation to exploit the error-free data on the target side, leading to more fluent outputs. Similarly, we exploit the error-free data on the target-side here - the original news sentences.

IV. DATASETS AND EXPERIMENTS

Although the system we build is completely unsupervised, we create a validation and test set for evaluating the performance of our model. We also use non-parallel news and tweet data. The detailed description and statistics of the datasets used is mentioned in this section. We conduct several experiments and perform ablation tests to highlight the importance of each module. In this section, we list the different experiments done and the experimental setups for all the experiments.

A. Datasets

To create the test and validation sets, we crawl disaster domain tweets using hashtags for different disaster events. We build a disaster domain classifier based on the work of Nguyen et al. [27]. This classifier was used to filter out the tweets that were relevant to the disaster domain. K-means clustering was used to cluster the tweets into four different topics. Clustering ensures that similar tweets are bunched together. We select one tweet from each of the clusters and give it to human annotators to create a news paragraph such that, it represents all the information in the tweet. Selection is done from different clusters to ensure diversity in tweets and avoid redundancy. Three annotators having graduate-level expertise in English language are used to create this dataset. The statistics of the curated dataset is given in Table I (News-Tweet-Parallel Validation and Test).

For training the unsupervised model, non-parallel tweet and news data were used. 22 million open domain tweets were crawled and news sentences were obtained from the WMT-2017 dataset. We use our domain classifier to filter out the disaster domain tweets from the open domain tweets. To obtain similar, in-domain news sentences from the WMT-2017 dataset, we use TF-IDF and cosine similarity between the filtered tweets and the news sentences. The data distribution for these non-parallel data is shown in Table I (Tweet-Non-Parallel and WMT-Non-Parallel).

The dataset we prepare to train XLM-MERGE was prepared by extracting propositions using ClauseIE [26] from these disaster domain news sentences. We drop all the sentences with fewer than two propositions. The concatenation of the propositions is limited to 512 BPE tokens, and longer sequences are truncated. The details of this dataset is also mentioned in Table I (News-Clause-Pair).

B. Experiments Conducted

We conduct the following experiments in this paper to better understand the significance of each step in the training process.

- **XLM-STY:** Since there are no previous models specific to this task, we use traditional unsupervised style transfer, using XLM as our baseline experiment (Discussed in Section III-B1). In this experiment we use the XLM-STY model and exclude the part 2 system i.e. XLM-MERGE. During evaluation, we pass four tweets concatenated together to the system and a news-style paragraph is thus obtained at the output.
- **XLM-STY-DIS + SYN:** In this experiment we use the XLM-STY-DIS model along with the synthetic-parallel-tweets training step. The XLM-MERGE model is not used here. We then pass the 4 tweets separately to the XLM-STY-DIS model and the outputs from XLM-STY-DIS model are concatenated to form the paragraph.
- **XLM-MERGE** In this experiment, four tweets are directly merged to form a paragraph using the XLM-MERGE model. There was no style transfer applied to the tweets.
- **XLM-STY + SYN + XLM-MERGE:** In this experiment, after obtaining the results from XLM-STY, XLM-MERGE is used to merge the 4 output sentences into a paragraph. We do not use a discriminator in training XLM-STY, but use the synthetic-parallel-tweets training step.
- **XLM-STY-DIS + XLM-MERGE:** In this experiment the result obtained from the XLM-STY-DIS experiment is merged into a paragraph using the XLM-MERGE model.

- **XLM-STY-DIS + SYN + XLM-MERGE:** In this experiment the outputs obtained from XLM-STY-DIS + SYN experiment is merged to form a paragraph using the XLM-MERGE model.

C. Experimental Setup

Lample et al. (2019) [22] demonstrated that initializing the encoder and decoder of a transformer network with their pre-trained model can significantly improve results for unsupervised machine translation. Since our task is to transfer style between news and tweets, we fine-tune an XLM-based Masked Language Model (MLM) with news and tweets as the two styles. For our initialization, we use the 15-language XLM-MLM model released by Lample et al. (2019) [22]. As mentioned in Section III-A, XLM uses only one shared encoder and one shared decoder for all the languages, while using different language-specific embeddings that get added to each token's embedding at the input. We initialize all of our model's parameters from theirs, and initialize the language-specific embeddings for news and tweets with the embedding corresponding to the English language from XLM. We then fine-tune our MLM model on the same Cloze task [28], as shown in IV-C. Following Lample et al., we randomly sample 15% of the input BPE tokens and replace them by a [MASK] token 80% of the time, by a random token 10% of the time while otherwise keeping them unchanged. During training, we use streams of 256 tokens, and mini-batch sizes of 8.

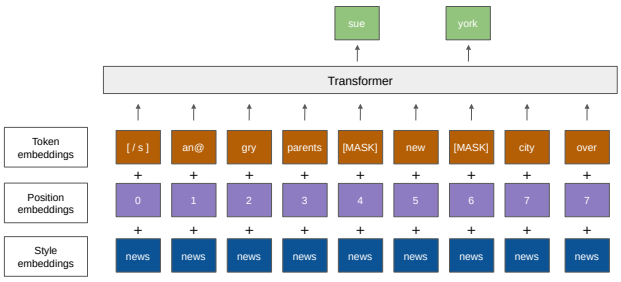

Fig. 2. XLM Masked Language Model (MLM) fine-tuning

In all our experiments for style-transfer and merging, we use a Transformer as our encoder-decoder model and initialize the encoders and decoders with our fine-tuned tweet-news XLM-MLM model. We use transformer models with 8 attention heads, 1024 hidden-units and GELU activations, and use 6-layers each for our encoders and decoders. We use the Adam optimizer with a learning rate of 1×10^{-5}, and mini-batches of size 4. Following Artetxe et al. (2018) [24], we back-translate each mini-batch on-the-fly using the model that is training itself. We also perform the adversarial-training and sythetic-parallel-data training steps on-the-fly in the same manner. During training, we alternate between the training objectives - L_{rec}, L_{bt}, L_D, L_{adv} and L_{syn}, with one-minibatch for each objective. For training XLM-STY and XLM-STY-DIS, we use the non-parallel WMT and Tweet datasets for training, and use the News-Tweet-Parallel data as validation and test. To train XLM-MERGE, we use the News-Clause-Pair dataset.

TABLE II
FIVE-POINT SCALES AND DEFINITION GUIDELINES FOR HUMAN ASSESSMENTS ON ADEQUACY AND FLUENCY

Metric	Score	Definition
Adequacy	1	None of the meaning is preserved
	2	Little of the meaning is preserved
	3	Much of the meaning is preserved
	4	Most of the meaning is preserved
	5	All the meaning is preserved
Fluency	1	Incomprehensible target sentence
	2	Dis-fluent target sentence
	3	Non-native kind of target sentence
	4	Good quality target sentence
	5	Flawless target sentence

During evaluation, if we pass all four style-converted tweets to the XLM-MERGE model together, the model would output a single sentence with all the information merged into one sentence. To avoid this, we pass the style-converted tweets two at a time to the XLM-MERGE model for merging.

V. RESULTS AND ANALYSIS

To evaluate the results of our experiments, we use both automatic and manual evaluation metrics. For automatic evaluation we use the tokenized BLEU scores computed using multi-BLEU pearl script included in Moses [2].

For manual evaluation we calculate adequacy and fluency of the generated text by following five-point scale system used by Rafael et al. [29]. While adequacy is the measure of source information (meaning) preservation at the output, fluency measures the quality of the target language constructions used in the translation. The grading system used for rating the adequacy and fluency of the output is detailed in Table II. The human evaluation was done by two human annotators with graduate level exposure. We randomly sampled 50 outputs from each experiment (300 in total), and distributed it to the annotators. They were asked to follow the grading scheme in Table II to score the outputs. We compute inter-annotator consistency using Fleiss' [30] kappa. We found the kappa of adequacy and fluency to be 0.74 and 0.76 respectively.

Results for the models in terms of BLEU scores is mentioned in Table III. Our baseline model 'XLM-STY' that uses vanilla XLM for style transfer gives a BLEU score of 14.34. Our final model 'XLM-STY-DIS + SYN + XLM-MERGE' outperforms the baseline in terms of BLEU (+4.98), adequacy (+0.83) and fluency (+0.96). The perfoemance gains were found to be statistically significant [3]. The model with best adequacy was found to be 'XLM-STY-DIS + SYN'. It is because this model's output is the input to the 'XLM-MERGE' model, therefore the error in terms of adequacy is propagated forward. Therefore the full model cannot have a better adequacy than this model. Another interesting observation from the results is that using only 'XLM-MERGE' on raw tweets produces good BLEU scores and fluency. However the adequacy is low since

[2]http://www.statmt.org/moses/

[3]Stastical significance t-test [31] was performed at 5% significance level

TABLE III
BLEU, ADEQUACY AND FLUENCY SCORES FOR THE DIFFERENT EXPERIMENTS CONDUCTED

Experiments	DIS	SYN	MERGE	BLEU	Adequacy	Fluency
XLM-STY	No	No	No	14.34	2.206	2.245
XLM-STY-DIS + SYN	Yes	Yes	No	18.28	**3.52**	2.34
XLM-MERGE	No	No	Yes	19.04	2.7275	2.63
XLM-STY-DIS + XLM-MERGE	Yes	No	Yes	16.31	2.74	2.7
XLM-STY + SYN + XLM-MERGE	No	Yes	Yes	19.04	3.04	3.02
XLM-STY-DIS + SYN + XLM-MERGE	Yes	Yes	Yes	**19.32**	3.04	**3.2**

the 'XLM-MERGE' tries to merge all the information, and treats noise in tweets as important information too.

The ablations clearly show the importance of each component of our model. In the experiment 'XLM-STY + SYN + XLM-MERGE' after removing the discriminator from the final model we see a reduction in the BLEU score by 0.28, and the fluency by 0.18. In the experiment 'XLM-STY-DIS + XLM-MERGE' the synthetic-parallel training step is removed, resulting in reduction in BLEU score by 0.28, adequacy by 0.30 and fluency by 0.5.

Some examples of the outputs produced by our model is shown below,

Tweet1: rain brings relief to dhaka as cyclone faniapproaches .
Tweet2: two people are reported to have died and more than a million have fled their homes after cyclonefani made landfall on india's east coast. more on this story here:.
Tweet3: thousands evacuated in eastern india as cyclone fani approaches.
Tweet4: 'extremely severe' cyclone fani to hit south of puri on friday: ndma.
OUT: rain brings relief to dhaka as two people are reported to have died and more than a million have fled their homes after biclonefani made landfall on india's east coast. more on this story . in eastern india as cyclone approaches , 's tremely severe .

Tweet1: deadly cyclone fani pummels india with wind gusts over 120 mph and flooding rain:.
Tweet2: cyclone fani wreaks havoc across east coast of india leaving three dead telegraph.
Tweet3: 3m people to evacuate as bangladesh braces for cyclonefani.
Tweet4: navys dornier aircrafts aerial photos show massive damage due to cyclone fani inpuri.
OUT: deadly cyclone pummels india with wind gusts over 120mph and flooding rain , leaving three dead telegraph . as bangladesh braces for biclonefani , navys

In the first example we can see that the model is able to merge the first two tweets fluently as they are both talking about relief and casualties. It makes a spelling mistake due to the noisy hash-tag. The third and fourth tweets contained more noise in the form of phrase 'more story on this'. This confused the model and it skipped the information in that sentence. In the second example too we can see the first sentence, that is obtained by joining two tweets, is fluent and complete. It successfully merges the tweets and gets rid of the redundant information. However the topic of the last two tweets were completely different. One tweet talks about evacuation and the other tweet is about aerial photographs. Such sentences cannot be joined, and hence our model fails to do so. Since we are randomly merging tweets, such issues persist.

Below are some examples of the outputs of XLM-MERGE, from the test data we prepared for the clause-merging task. These outputs show that XLM-MERGE is sufficiently robust, and learns to produce succinct sentences when provided one or more clauses with redundant information. This reiterates that the outputs of our whole model can be improved if the style-transfer step produces less noisy outputs from the tweets.

IN: telegraph media group helped fund. telegraph media group helped the project. telegraph media group owns the daily telegraph. the daily telegraph is daily.
OUT: telegraph media group, which owns the daily telegraph, helped fund the project.

IN: he will participate in seminars and on-campus activities. he will participate throughout the year.
OUT: he will participate in seminars and on-campus activities throughout the year.

VI. CONCLUSION AND FUTURE WORK

In this paper, we explore a new task of creating a news paragraph from a given set of tweets in the disaster domain in an unsupervised manner. We intend this to be an initial step towards solving the problem of unsupervised guided text generation, where we intend to control the style and the content of generated text. We propose a pipeline system with two models to tackle this problem. Our first model transfers the style of informal and noisy tweets into more formal news-like sentences in an unsupervised setting. Our second model solves the problem of merging together several information nuggets across sentences to form a single paragraph. We also prepare and release an evaluation dataset for the task. We propose novel methods to train both our style transfer and merging models, and show through automated and human evaluations

that our proposed methodology can outperform the baseline style transfer models.

In the future, we would like to explore better merging strategies for sentences, where the model learns when to merge and when to avoid merging sentences. We would also like to explore an end-to-end strategy to achieve merging and style transfer of sentences in one go, to avoid the propagation of error from .

ACKNOWLEDGMENT

The research reported in this paper is an outcome of the project titled "A Platform for Cross-lingual and Multilingual Event Monitoring in Indian Languages", supported by IMPRINT-1, MHRD, Govt. of India, and MeiTY, Govt. of India.

REFERENCES

[1] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1700–1709.

[2] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," *arXiv preprint arXiv:1710.11041*, 2017.

[6] Z. Zhang, S. Ren, S. Liu, J. Wang, P. Chen, M. Li, M. Zhou, and E. Chen, "Style transfer as unsupervised machine translation," *arXiv preprint arXiv:1808.07894*, 2018.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[8] Z. Yang, Z. Hu, C. Dyer, E. P. Xing, and T. Berg-Kirkpatrick, "Unsupervised text style transfer using language models as discriminators," in *Advances in Neural Information Processing Systems*, 2018, pp. 7287–7298.

[9] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: A simple approach to sentiment and style transfer," *arXiv preprint arXiv:1804.06437*, 2018.

[10] A. Al Nahas, M. S. Tunali, and Y. S. Akgul, "Supervised text style transfer using neural machine translation: Converting between old and modern turkish as an example," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2019, pp. 1–4.

[11] M. Shang, P. Li, Z. Fu, L. Bing, D. Zhao, S. Shi, and R. Yan, "Semi-supervised text style transfer: Cross projection in latent space," *arXiv preprint arXiv:1909.11493*, 2019.

[12] Z. Jin, D. Jin, J. Mueller, N. Matthews, and E. Santus, "Unsupervised text style transfer via iterative matching and translation," *arXiv preprint arXiv:1901.11333*, 2019.

[13] F. Luo, P. Li, J. Zhou, P. Yang, B. Chang, Z. Sui, and X. Sun, "A dual reinforcement learning framework for unsupervised text style transfer," *arXiv preprint arXiv:1905.10060*, 2019.

[14] M. Litvak, J. Conroy, and P. A. Rankel, "Ranlp 2019 multilingual headline generation task overview," in *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, 2019, pp. 1–5.

[15] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," *arXiv preprint arXiv:1708.00625*, 2017.

[16] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[17] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1171–1181.

[18] F. Kiyomarsi and F. R. Esfahani, "Optimizing persian text summarization based on fuzzy logic approach," in *2011 International Conference on Intelligent Building and Management*, 2011.

[19] F. Chen, K. Han, and G. Chen, "An approach to sentence-selection-based text summarization," in *2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM'02. Proceedings.*, vol. 1. IEEE, 2002, pp. 489–493.

[20] Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarization using wikipedia," *Information Processing & Management*, vol. 50, no. 3, pp. 443–461, 2014.

[21] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields." in *IJCAI*, vol. 7, 2007, pp. 2862–2867.

[22] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *Advances in Neural Information Processing Systems*, 2019, pp. 7057–7067.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," *arXiv preprint arXiv:1710.11041*, 2017.

[25] M. Kaufmann and J. Kalita, "Syntactic normalization of twitter messages," in *International conference on natural language processing, Kharagpur, India*, 2010.

[26] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 355–366.

[27] D. T. Nguyen, K. A. Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, "Robust classification of crisis-related data on social networks using convolutional neural networks," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[28] W. L. Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism quarterly*, vol. 30, no. 4, pp. 415–433, 1953.

[29] R. E. Banchs, L. F. D'Haro, and H. Li, "Adequacy–fluency metrics: Evaluating mt in the continuous space model framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 472–482, 2015.

[30] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[31] B. L. Welch, "The generalization ofstudent's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.