Generative Adversarial Network for Joint Headline and Summary Generation

CHING-SHENG LIN¹, JUNG-SING JWO¹,², CHENG-HSIUNG LEE¹, AND TSAI-FENG HSIEH¹
¹Master Program of Digital Innovation, Tunghai University, Taichung 40704, Taiwan
²Department of Computer Science, Tunghai University, Taichung 40704, Taiwan
Corresponding author: Ching-Sheng Lin (cslin612@thu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2222-E-029-002.

ABSTRACT With the ever-increasing amount of electronic documents being generated, it is imperative to provide an intuitive headline and a concise summary of the document to help readers quickly get the gist without going through the all details. While humans have strong abilities to create headlines and generate summaries, the automatic text generation of this research field is still challenging due to the difficult language understanding and complex text synthesis. Moreover, human annotation for machine learning is another matter needed to be addressed. In this paper, we propose a joint model to resolve aforementioned issues simultaneously. To perform unsupervised training to reduce labor cost, we employ a generative adversarial network (GAN) without parallel training data. Considering that the headline and summary have strong relationship, we propose a joint model to learn better representations by including an additional type representation into the GAN. Experiments are conducted on the public dataset, NEWS-ROOM. The experimental results demonstrate that our approach is able to effectively create a reasonable headline as well as a concise summary.

INDEX TERMS Headline, summary, joint model, unsupervised training, generative adversarial network.

I. INTRODUCTION

Due to the rapid development of information and communication technologies, there are tremendous amount of text data generated all the time from various sources. A succinct and informative representation can not only save readers’ time, but also increase the usage of salient information for the downstream tasks. It is, therefore, imperative to provide a convenient and fast tool for people to digest the information. Data representation learning and text generation are popular topics in the NLP community. With the recent advancement of deep learning techniques, the sequence-to-sequence auto-encoder (SA) [1] and generative adversarial network (GAN) [2] are two important neural network architectures commonly used for NLP tasks. SA is proposed for the latent representation using a recurrent neural network (RNN), known as an encoder, to encode the input text as a low-dimensional vector representation followed by another RNN, known as a decoder, to reconstruct the original input. Since the compressed vector representation is viewed as an act of communication between the encoder and decoder, the content is usually not easy for human to comprehend. To enhance the readability of the representation, the GAN model has been adopted to address the challenges where the generator produces the representation based on the encoder and the discriminator forces the representation needed to be human-readable [3], [4].

Automatic text summarization is a task of creating an informative, coherent and shorter version of the original text while preserving its main gist. It is expected to concatenate several sentences from the original document or rewrite an abstract to summarize the whole document where the former is an extractive method and the latter is an abstractive method. Recently, access to the internet for the information search through handheld devices is growing significantly. However, a lengthy summary is not suitable to be displayed on mobile phones for users to glance. Automatic headline generation aims at summarizing the vital points of the document within one sentence and is an applicable solution to deal with the problem [5], [6], [7]. Headlines and summaries are used to
represent the main ideas of the articles with different levels of compression.

Joint training methods have been proposed to model the dependencies between different tasks and improve the performance against independent models simultaneously. For NLP problems, a set of shared linguistic layers for different tasks would benefit individual tasks by training a joint model [8]. Since the headline and summary generation are not two independent tasks, they can mutually enhance each other by jointly optimizing the objectives of two tasks. In this paper, we establish an automatic generation system for headline and summary. With the development of the proposed system, we can reduce the burden of reading lengthy documents and improve the efficiency for readers. Our end-to-end network is based on the GAN model which works in an unsupervised manner and does not require human annotations. The overview of our GAN for joint headline and summary generation is shown in Figure 1. It is consisted of three components: one is a generator module, a second is a discriminator and a third is a reconstructor. Both generator and discriminator take a sequence concatenated with a type vector which is related to a headline or summary as the input. The output of generator is a sequence of word distributions and discriminator, on the other hand, outputs a scalar to distinguish between human-written and machine-generated text. The purpose of reconstructor is to reconstruct the source input from the output of generator.

The contributions of this paper lie in three folds:
- We propose an end-to-end model based on the GAN to generate headlines and summaries simultaneously. A generator is used to produce headlines and summaries, and a discriminator is designed to evaluate the qualities of the generated texts.
- Our approach operates in an unsupervised setting and does not require paired training data. To the best of our knowledge, it is the first unsupervised end-to-end model which is able to create different granularities of summaries at the same time.
- Experimental results on the NEWSROOM dataset show that our approach is able to achieve encouraging performance.

The remainder of this paper is organized as follows. Section 2 discusses several research efforts relevant to the work of this paper. We explain the proposed network and techniques in Section 3. Empirical studies are given to illustrate the effectiveness of the methodology in Section 4. In section 5, we present some conclusions and identify several possible future directions.

II. RELATED WORK

In this section, we describe some key research topics related to this paper including text summarization, headline generation, joint training and GAN.

The automatic text summarization can be generally categorized into extractive and abstractive methods. Extraction based approaches take the sentences from the original document as candidates to compose a summary whereas abstraction based methods aim at understanding the main information within the document to rewrite a shorter content. With the advent of deep neural network, many attempts have been made for these two research directions. To overcome the redundant phrases in the selected sentences of extractive summarizations, a hierarchical attentive heterogeneous graph
approach is proposed to model redundancy dependencies and measure salience between sentences simultaneously [9]. The Transformer is a sequence-to-sequence (seq2seq) architecture based on attention mechanism and has achieved great successes in many NLP applications [10]. For the unsupervised extractive summarization, STAS model pre-trains for a masked sentence prediction and the order recovery of shuffled sentences based on a hierarchical transformer [11]. The attention weights of the pre-trained hierarchical transformer are then used to calculate the importance of each sentence and the top sentences are taken to constitute the summary. Recently, there has been an increasing interest in the area of unsupervised abstractive summarization models. SEQ3 consists of two chained auto-encoders where the first auto-encoder (Compressor) generates a summary from the input text and the second auto-encoder (Reconstructor) attempts to reproduce the input [12]. In order to encourage the Compressor to generate human-readable text, a pre-trained language model on the full training data acts as a prior. TED is an unsupervised abstractive summarization system based on Transformer and pre-trains on unlabeled corpora by predicting the first three sentences of the document [13]. It is then fine-tuned with theme modeling and denoising auto-encoder. Theme modeling forces the summary semantically close to the input source and denoising auto-encoder prevents the summary from simply copying the input con-text.

Compared with the text summarization, the headline is a much more condensed representation of the input document. Since the headline generation for long news articles requires strong reasoning abilities, Universal Transformer architecture [14] is applied to non-local representations of the text and achieves state-of-the-art results on the New York Times Annotated Corpus [15]. A coarse-to-fine neural model first extracts the important sentences of a document using the existing extractive summarization techniques, and then applies seq2seq model to process the important sentences and produces the headline [16]. One of important applications in headline generation is for product title creation. To produce short descriptions for the E-commerce products, a feature-enriched neural extractive model combines three levels of features including Content, Attention and Semantic parts [17]. The model formulates the title generation as a RNN-based sequence tagging problem by predicting whether a given word should be chosen into the summary.

Joint training is a commonly used strategy when related tasks can be trained together and share similar data [18]. Regarding joint training in the neural network model, since tasks share a certain number of model layers, this encourages the model to learn more generalizable representations and maximizes the generalization performance on all tasks. Joint training has been shown to accomplish promising results in different fields of NLP. Named entity recognition (NER) and relation extraction (RE) are two crucial and related tasks in Information Extraction (IE). A new tagging scheme is proposed to assign each word to a hybrid tag which contains both entity and relation types. With the tagging scheme, the entities and relations can be extracted using the same network [19]. Slot filling (SF) and intent detection (ID) are two import subtasks in spoken language understanding (SLU). The objective of SF is to obtain necessary information for each token in the utterance and ID aims to identify the goal of the user utterance. Relying on the correlations of the two tasks, Gated recurrent unit (GRU) is used to predict the slot label by learning the feature of each time step and classify the intent by capturing global features of the given utterance [20].

GAN framework is introduced to train a generator and a discriminator through an adversarial process where the generator creates synthetic data and the discriminator determines whether a sample is from the real data or generated by the generator [2]. It was first applied in image generation and has shown considerable success in many fields including natural language generation. However, directly applying GAN to generate text may lead to training difficulties if words are sampled by non-differentiable operators such as argmax function. The Wasserstein GAN (WGAN) is adopted to directly estimate the Wasserstein-1 distance (also known as Earth-Mover distance) between distributions from real and generated samples [3], [21]. Reinforcement learning based approach is proposed to address the non-differentiable issue where the generator plays the role of an agent to produce the summary and the discriminator is used as a reward function to distinguish the human-written summary from machine-generated summary [22]. ConGAN which is an end-to-end model directly outputs the embeddings of each word rather than the probability distribution over the vocabulary, making the training process differentiable and more stable [23].

III. PROPOSED METHOD

In this section, we will present the proposed joint network, which contains a generator G with parameter \( \theta_g \), a discriminator D with parameter \( \theta_d \), and a reconstructor R with parameter \( \theta_r \). The joint optimization strategy will be discussed as well.

A. THE NETWORK ARCHITECTURE

Our model shown in Figure 2 targets at generating the headline and summary for the given source text. The mathematical notations and functions are described below:

- The i-th source document is represented by \( x_i \) which is consisted of words as \( \{x_{i,1}, x_{i,2}, \ldots\} \).
- A type vector \( T \) is used to represent a headline (e.g., \( t=1 \)) or summary (e.g., \( t=2 \)).
- The G accepts a source text \( x_i \) and produces an output \( \hat{y}^T_i \) which is consisted of sampled words as \( \{\hat{y}^T_{i,1}, \hat{y}^T_{i,2}, \ldots\} \).
- The R takes \( \hat{y}^T_i \) and tries to reconstruct back to \( x_i \).
- The D accepts a text and type vector \( T \) to output a scalar.

The G which is a seq2seq model takes as an input a source document \( x_i \). We adopt the hybrid pointer-generator network [24] to directly copy words from the input text or instead generate words from the predefined vocabulary. The
encoder vector representation $V_G$ of the G is combined with a type vector $T^i$ to form the input of the decoder. For the decoder of the G, at each time step, we apply sampling techniques to obtain the final output $\hat{y}_i^1$ which represents a headline or summary.

Keeping the most salient information of source text is a major concern while evaluating the performance of the generated results. However, without imposing any constraints on G, it may produce text unrelated to the source document. In this paper, we develop a reconstructor R by taking the output from G as the input and reconstructing to the original text $x^i$ by an encoder vector representation $V_R$. In this way, R will enforce G to contain sufficient information in order to achieve the objective of reconstruction. The training goal of R is to make the sampled text $\hat{x}_i$ as close to input $x_i$ as possible.

In addition to the generator, there should be a component to evaluate the generated text of G and distinguish between the real text and generated results. We adopt a discriminator D to ensure that G will produce the headline and summary which hold good quality and should be human-readable. The structure of D is a binary classifier which consists of an LSTM layer. The LSTM layer takes as an input a text with a type structure of $x^i$. In this way, the generated text $\hat{y}_i^1$ is used to reconstruct the input sentence $x_i$ from the generated text $\hat{y}_i^1$. The reconstruction loss is defined as the following:

$$Loss_D = \frac{1}{m} \sum_{i=1}^{m} \log D(T^1, y_i^1) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(T^1, G(T^1, x_i))) + \frac{1}{m} \sum_{i=1}^{m} \log D(T^2, y_i^2) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(T^2, G(T^2, x_i)))$$

where G ($T^1$, $x_i$) is $\hat{y}_i^1$ and G ($T^2$, $x_i$) is $\hat{y}_i^2$.

To keep the information integrity and learn the mutual information with the source input text, the purpose of R is used to reconstruct the input sentence $x_i$ from the generated text $\hat{y}_i^1$. The reconstruction loss is defined as the following:

$$Loss_R = \frac{1}{m} \sum_{i=1}^{m} -\log P_R(x_i | G(T^1, x_i)) + \frac{1}{m} \sum_{i=1}^{m} -\log P_R(x_i | G(T^2, x_i))$$

By minimizing the above loss function, the G will be forced to produce output text related to the input document.

In our model, the G and R form an auto-encoder structure whereas the G and D compose a generative adversarial network. In the adversarial process, the G and D are trained together in a competitive manner. The positive training samples of the D are from human-written headlines ($y_i^1$) and summaries ($y_i^2$) while the negative samples are the generated headlines ($\hat{y}_i^1$) or summaries ($\hat{y}_i^2$) by the G. The LSTM model of D takes a discrete word sequence as input and predicts the current score $s_j$ based on $\{y_{k,1}^1, y_{k,2}^1, \ldots, y_{k,j}^1\}$. The output of D is the average of the summation of $\{s_1, s_2, \ldots\}$. The loss function of the D is defined as:

$$Loss_D = \frac{1}{m} \sum_{i=1}^{m} \log D(T^1, y_i^1) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(T^1, G(T^1, x_i))) + \frac{1}{m} \sum_{i=1}^{m} \log D(T^2, y_i^2) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(T^2, G(T^2, x_i)))$$

The G receives source input $x_i$ and produces the related text $\hat{y}_i^1$ based on $T^i$. The $\hat{y}_i^1$ and $T^i$ form a paired output which will be sent to the D. Since the D is an LSTM layer, we predict the next word at each time step by choosing the maximum probability from the output distribution. In order to train...
the G, the paired output is fed to D and the corresponding loss of the D is back-propagated to update weights. However, due to the sampling process of the LSTM layer, the above training will suffer from the non-differentiable issue. We use policy gradient approach to resolve this issue [3], [22]. The G plays the role of an actor to choose the next word from the vocabulary of words and obtains rewards from the D to encourage the G performing better. The loss function of the G is defined as:

$$\text{Loss}_G = \frac{1}{m} \sum_{i=1}^{m} \log D(T^1, G(T^1, x_i)) + \frac{1}{m} \sum_{i=1}^{m} \log D(T^2, G(T^2, x_i))$$ (3)

The training pseudo code of our model is illustrated in Algorithm 1. Firstly, we draw examples of input text $x_i$ (lines 2), headlines $y^1_i$ and summaries $y^2_i$ (lines 3). Note that $x_i$ is not paired with $y^1_i$ and $y^2_i$. The training of the D is in line 5-6 based on the gradient of $\text{Loss}_D$ with respect to the $\theta_d$. After optimizing the D, we train the R (line 8-9) with $\text{Loss}_R$ in Eq. (2). Similar to the D and R, the training of G is in line 11-12 by the gradient of $\text{Loss}_G$, $\alpha_d$, $\alpha_r$, and $\alpha_g$ are the learning rates for the D, R and G respectively.

Algorithm 1  GAN for Joint Headline and Summary Generation

Parameters: $\theta_d$ for the discriminator D, $\theta_r$ for the reconstructor R and $\theta_g$ for the generator G.

1. for each training iteration:
2. Sample $m$ examples of source text $x_1, x_2, \ldots, x_m$
3. Sample $m$ examples of headlines $y^1_1, y^1_2, \ldots, y^1_m$ and $m$ examples of summaries $y^2_1, y^2_2, \ldots, y^2_m$
4. # Update D to maximize $\text{Loss}_D$
5. Compute the gradient of $\text{Loss}_D$ in Eq. (1) to obtain $\nabla_{\theta_d} \text{Loss}_D$
6. Update $\theta_d$ using $\theta_d + \alpha_d \times \nabla_{\theta_d} \text{Loss}_D$
7. # Update R to maximize $\text{Loss}_D$
8. Compute the gradient of $\text{Loss}_R$ in Eq. (2) to obtain $\nabla_{\theta_r} \text{Loss}_R$
9. Update $\theta_r$ using $\theta_r + \alpha_r \times \nabla_{\theta_r} \text{Loss}_R$
10. # Update G to maximize $\text{Loss}_G$
11. Compute the gradient of $\text{Loss}_G$ in Eq. (3) to obtain $\nabla_{\theta_g} \text{Loss}_G$
12. Update $\theta_g$ using $\theta_g + \alpha_g \times \nabla_{\theta_g} \text{Loss}_G$
13. end for

IV. EXPERIMENTS AND RESULTS

In this section, we conduct experiments to evaluate the proposed method for headline and summary generation consisting of: (1) the experimental dataset; (2) evaluation indicators; (3) comparisons with other approaches; and (4) ablation studies to examine the strengths and weaknesses of our model.

A. DATASET

The Cornell Newsroom is a large and public dataset with 1.3 million articles from 38 news publishers extracted between 1998 and 2017 [26]. The data are stored in JSON format which contains the headline, context and human-written summary. In general, the average length of the article is about 658 words, the mean headline size is around 9.55 words and the average size of the summary is 31.32 words. We choose this dataset for the end-to-end test since it contains both headline and summary for each article.

B. EVALUATION METRICS

To measure the performance of the proposed approach, we employ ROUGE metrics which are commonly used to check the similarity between machine-generated text and human-written text. In the research field of summarization, there are three well-known ROUGE variants which are ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L). R-1 compares the unigram overlap between the generated summary and the ground-truth summary while R-2 measures the overlap of bigrams. Similarly, R-L computes the longest common sequence between generated and ground-truth summary to assess fluency.

C. PERFORMANCE RESULTS

We compare our results against several baselines on NEWS-ROOM dataset. LEAD-X which is a simple baseline uses the top X words as a result where the value of X is 10 for the headline task and 30 for the summary generation in this experiment. SHEG is proposed to produce both an abstractive summary and a headline in a supervised manner by selecting salient phrases and combing a pointer-generator network with a controlled actor-critic model [27]. Pointer-Generator with Coverage (PGC) is another supervised model based on a hybrid sequence-to-sequence attentional model [24]. For the unsupervised baselines, Summary Loop [28] and Adversarial REINFORCE-based GAN (AdvREGEN) [3] are two methods chosen to compare with our model. Note that SHED is the only model dedicated for the generation of headlines and summaries but with supervised method. The hyper-parameter setting of the model is displayed in Table 1. The experimental comparison is carried out on the Windows system with an 11th Gen Intel(R) Core(TM) i9-11900K@3.50GHz, NVIDIA GeForce RTX 3090 GPU 24G and 128 GB RAM.

From the comparison results in Table 2, several phenomena are observed. First, in the headline generation, our approach achieves the best performance on R-1 and R-L scores. Compared to the second best result, our method shows an improvement of 1.11% on R-1 and 0.29% on R-L. In particular, it even surpasses the performances of supervised method, SHEG. Second, in the summary generation, our model obtains the highest R-1 score (29.51%). Third, our method yields slightly lower R-2 score indicating our approach can extract important concepts of the news but has low rate of bigrams overlap with ground-truth summaries. Fourth, the supervised approaches do not fully dominate the Rouge scores but have the
C.-S. Lin et al.: Generative Adversarial Network for Joint Headline and Summary Generation

V. CONCLUSION
In this study, we present an unsupervised method based on a generative adversarial network to jointly address the problem of the headline and summary generation. The experiment is conducted on the NEWSROOM dataset and shows encouraging results. Compared with other approaches including supervised and unsupervised learning methods, our model achieves the best scores in terms of ROUGE-1 and ROUGE-L for the headline generation, and the best ROUGE-1 score for the summary generation.

As our model has lower ROUGE-2 score suggesting that it gains better unigram performance at the cost of low bigram results, the future work will be to improve the ROUGE-2 performance. Including the language model as another discriminator to our network would be helpful to reduce the noise, thereby yielding better summaries. Another possible future work might use contextual representation such as BERT to further improve the generation results.

D. ABLATION STUDIES
In the end, to evaluate the individual contribution of the component in the proposed model, we perform ablation experiments to make further analysis and display the results in Table 4. When we train two models for headline and summary generation respectively without joint optimization, all the Rouge scores become lower. This result validates the effectiveness of the joint training method.

TABLE 1. Hyper-parameter configuration.

| Component   | Parameter       | Value   |
|-------------|-----------------|---------|
| General     | batch size      | 8       |
| Generator   | LSTM hidden units | 300     |
|             | embeddings size | 300     |
|             | learning rate   | 0.00005 |
| Discriminator | LSTM hidden units | 500     |
|             | embeddings size | 300     |
|             | learning rate   | 0.002   |
| Reconstructor | LSTM hidden units | 200     |
|             | embeddings size | 100     |
|             | learning rate   | 0.0001  |

TABLE 2. Performance comparison (%) where we use '-' if the results are not available.

|          | Headline  | Summary  |
|----------|-----------|----------|
|          | R-1 | R-2 | R-L | R-1 | R-2 | R-L |
| Lead-X   | 5.39 | 0.60 | 5.04 | 13.78 | 3.00 | 11.92 |
| SHEG     | 13.81 | 7.94 | 11.42 | 28.2 | 14.7 | 25.9 |
| PGC      | -    | -    | -    | 27.5 | 13.3 | 23.5 |
| Summary  | -    | -    | -    | 27.0 | 9.6  | 26.4 |
| AdvRGM   | 14.53 | 0.09 | 12.18 | 22.99 | 2.83 | 17.29 |
| Our Model| 15.64 | 1.64 | 12.47 | 29.51 | 6.43 | 22.35 |

TABLE 3. Example output produced by our model.

| DOCUMENT: | big show will continue to put on a show that includes rope climbing . . . sources connected with the wwe tell tmz . john cena averted serious injury during a match in manila friday when the rope on which big show was standing snapped , sending him tumbling to the mat . cena rolled to avoid impact . we contacted wwe sources to find out if big show will continue standing on the ring ropes . if his 440 pound weight was simply too much , the wwe could face serious liability if it happened again and someone were injured . we re told the accident was a fluke and the rope was simply defective . our sources say the ropes are big show-worthy and the wrestler will go into the ring in future matches undeterred |
| GOLD HEADLINE: | big show will be back on the ropes |
| GOLD SUMMARY: | big show will continue to put on a show that includes rope climbing despite almost causing serious injury to john cena |
| GEN HEADLINE: | big show will be on rope that includes climbing |
| GEN SUMMARY: | big show will be continue to a that includes rope climbing . john serious injury match in manila |

TABLE 4. Ablation experimental results (%).

|          | Headline  | Summary  |
|----------|-----------|----------|
|          | R-1 | R-2 | R-L | R-1 | R-2 | R-L |
| Our Model| 15.64 | 1.64 | 12.47 | 29.51 | 6.43 | 22.35 |
| W/o Joint| 12.74 | 0.09 | 11.09 | 27.40 | 3.20 | 19.90 |
