Information Sciences 521 (2020) 46–61

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint

Min Yang[a], Xintong Wang[b], Yao Lu[c], Jianming Lv[b], Ying Shen[d], Chengming Li[a,*]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[b] School of Computer Science and Engineering, South China University of Technology, China
[c] School of Computer Science, University of Waterloo, Canada
[d] School of Electronics and Computer Engineering Peking University Shenzhen Graduate School, China

ARTICLE INFO

Article history:
Received 1 February 2019
Revised 24 December 2019
Accepted 12 February 2020

Keywords:
Abstractive text summarization
Generative adversarial network
Multi-task learning

ABSTRACT

Abstractive text summarization is an essential task in natural language processing, which aims to generate concise and condensed summaries retaining the salient information of the input document. Despite the progress of previous work, generating summaries, which are informative, grammatically correct and diverse, remains challenging in practice. In this paper, we present a Plausibility-promoting Generative Adversarial Network for Abstractive Text Summarization with Multi-Task constraint (PGAN-ATSMT), which shows promising performance for generating informative, grammatically correct, and novel summaries. First, PGAN-ATSMT adopts a plausibility-promoting generative adversarial network, which jointly trains a discriminative model D and a generative model G via adversarial learning. The generative model G employs the sequence-to-sequence architecture as its backbone, taking as input the original text and generating a corresponding summary. A novel language model based discriminator D is proposed to distinguish the generated summaries by G from the ground truth summaries without the saturation issue in the previous binary classifier discriminator. The generative model G and the discriminative model D are learned with a minimax two-player game, thus this adversarial process can eventually adjust G to produce high-quality and plausible summaries. Second, we propose two extended regularizations for the generative model G using the multi-task learning, sharing its LSTM encoder and LSTM decoder with text categorization task and syntax annotation task, respectively. The auxiliary tasks help to improve the quality of locating salient information of a document and generate high-quality summaries from language modeling perspective alleviating the issues of incomplete sentences and duplicated words. Experimental results on two benchmark datasets illustrate that PGAN-ATSMT achieves better performance than the state-of-the-art baseline methods in terms of both quantitative and qualitative evaluations.

* Corresponding author.
E-mail addresses: w.xintong@mail.scut.edu.cn (X. Wang), luyao@uwaterloo.ca (Y. Lu), jmlv@scut.edu.cn (J. Lv), shenying@pkusz.edu.cn (Y. Shen), cm.li@siat.ac.cn (C. Li).

https://doi.org/10.1016/j.ins.2020.02.040
0020-0255/

1. Introduction

Artificial intelligence studies have witnessed great interests in generating concise summaries automatically that retain the salient information of the input text, which is known as the abstractive text summarization. As opposed to extractive text summarization, which identifies the best summarizing sentences from the input text, abstractive text summarization models can generate summaries containing new words and phrases that do not appear in the original text. Recently, abstractive text summarization has attracted much attention due to its board applications for information condensation.

Motivated by the remarkable progress of the encoder-decoder framework in dialogue generation, neural machine translation and image captioning, most abstractive text summarization systems adopt the sequence-to-sequence (seq2seq) method to produce summaries [31,36]. The general idea of the seq2seq framework is to use a long short-term memory (LSTM) network to encode the input text and then feed the representation vector to an LSTM decoder to generate summaries. These seq2seq based approaches have become the mainstream due to their capability of capturing the syntactic and semantic relations between raw texts and summarises in an end-to-end way. Despite the significant success of existing abstractive text summarization systems, generating summarises that are accurate, concise, and fluent remains a challenge for several reasons.

First, based on our empirical observation, summary styles for different text categories can vary significantly. As demonstrated in Table 1, two common categories (i.e., *Sports* and *Politics*) in CNN/Daily Mail Corpus [16] are taken as an example. To summarize a politic event, people have a tendency to emphasize the subject of the event, and the result or influence of the event. In contrast, a sport summary is expected to include the teams and scores of the sport event. Obviously, the generated summaries should pay particular attention to different topics which belong to the corresponding categories. However, most previous approaches [22,33] employ a uniform model to produce summaries for the source documents from different categories, which are prone to generate generic and trivial summaries that easily miss or under-represent important aspects of the original documents. Furthermore, in an ablation study of our model (see Table 2), a model which does not recognize the text category could generate a descriptive summary missing a salient entity in the text.

Second, syntactic information plays a crucial role in sentence generation [30]. Enforcing syntactic conformance addresses issues like incomplete sentences and duplicated words. As shown in Table 2, the model which is unaware of the text syntax could generate a broken summary for the given document. Yet, an improved system whose component has been co-trained with syntax annotation task could generate a more satisfied sentence for accurately and correctly condensing the raw document. Despite its usefulness, syntax information is underutilized in abstractive text summarization.

Third, in existing studies, the seq2seq based methods are usually trained to generate summaries of input documents via the maximum likelihood estimation (MLE) algorithm. Nevertheless, the MLE based models have two major disadvantages. (i) The evaluation metrics used in testing are not used when training the model, which will magnify the differences between losses from training and testing. For example, the seq2seq models are typically trained by employing the cross-entropy strategy, while they are evaluated by employing non-differentiable and discrete evaluation metrics such as ROUGE [21] at test time. (ii) While training, the decoder often receives the word vector of the previous ground-truth word at each time step. However, at testing phase, the decoder takes as input the previous word emitted by the seq2seq model, which may result in the exposure bias issue [34]. which may accumulate errors quickly at each time step. Specifically, when the de-

Table 1
Two example articles and their summaries from *politics* and *sports* categories, respectively.

Category: Politics
Article (truncated): "isis claimed it controlled part of iraq's largest oil refinery sunday, posting images online that purported to show the storming of the facility, fierce clashes and plumes of smoke rising above the contested site. the group said it launched an assault on the baiji oil refinery late saturday. by sunday, isis said its fighters were inside the refinery and controlled several buildings, but iraqi government security officials denied that claim and insisted iraqi forces remain in full control. cnn couldn't independently verify isis' claim. it wouldn't be the first time that militants and iraqi forces have battled over the refinery, a key strategic resource that has long been a lucrative target because the facility refines much of the fuel used by iraqis domestically. if an attack damaged oil fields or machinery, it could have a significant impact. the refinery is just 40 km (25 miles) from the northern iraqi city of tikrit, which iraqi forces and shiite militias wrested from isis less than two weeks ago. cnn's jennifer deaton and catherine. shoichet contributed to this report."
Reference summary: "isis says it controls several buildings at the baiji oil refinery. iraqi government security officials say iraqi forces remain in full control the refinery, iraq 's largest, has long been a lucrative target for militants."
Category: Sports
Article (truncated): "Article (truncated): serena williams claimed her eighth miami open title in 14 years after ruthlessly brushing aside the challenge of 12th seed carla suarez navarro in saturday's final. the world no 1 won the final 10 games in a 6-2 6-0 demolition of her spanish opponent to claim her third straight title at an event she has dominated since winning her first crown back in 2002. serena williams poses on the beach with the championship trophy after defeating carla suarez navarro. williams poses with the road to singapore sign post on crandon park beach after her straight-sets victory. (... ...) williams saved a break point in her opening service game and then broke to love in the next to leave suarez navarro with a mountain to climb. there would be no way back, with suarez navarro winning just two points on serve in the second set. williams, meanwhile, won 21 of 22 points on her first serve. spain's suarez navarro started strongly but was no match for serena williams. williams once again broke to love to move 5-0 ahead before clinically wrapping up the match inside 57 minutes. suarez navarro was quick to hail williams, adding in quotes broadcast by bt sport 1: all that you have, you deserve and for me you are the number one right now."
Reference summary: "serena williams won her eighth miami open title on saturday. she won the final 10 games in a 6-2 6-0 demolition in miami. the unbeaten world no 1 has now won 12 consecutive finals."

Table 2
An example of article from *Show* category and its summaries by different models. The words in red indicate the incomplete or redundant phrases.

Category: Show
Article (truncated):"they are one of the world's most famous couples-and have quickly gained respect among the fashion elite. and now, one esteemed designer has revealed why kim kardashian and kanye west have the midas touch. olivier rousteing has revealed that he chose kim and kanye to star in balmain's latest campaign because they 'represent a family for the new world'. scroll down for video. fashion's most well-connected designer, olivier rousteing, has revealed why he snapped kim kardashian and kanye west up to front his balmain campaign. (... ...) the 29-year-old creative director says he snapped up the duo, seen wearing his designs, because they are among the most talked-about people and embody the idea of a modern family. olivier-who regularly dresses kim, 34, and her siblings for the red carpet-explained that when kendall jenner and kim wear his clothes, they look like a 'fashion army'. the whole family seem enamoured with rousteing's designs and kim and kanye often sport matching outfits by the french fashion house. kim and kanye this week made trips to france and armenia with their daughter, north west. the trip to the religious mecca reportedly included north being baptised in the country where her late father's side of the family originated from. kim kardashian, kanye west and north visit the geghard monastery in armenia and take in the sights. kim, kanye and north have become a fashionable family. pictured here with alia wang, aimie wang and nicki minaj at the alexander wang show in february2014.//
Reference summary: "olivier rousteing has revealed why he chose kim and kanye for balmain. designer says the couple are among the most talked-about people. fashionable couple love wearing matching designs by balmain designer."
Summary by our model: "olivier rousteing has revealed why he chose kim and kanye to star in balmain's latest campaign because they represent a family for the new world. french designer says the couple are among the most talked-about people."
Summary by our model without text categorization: "olivier rousteing has seen kim kardashian and kanye west. kim kardashian and kanye west have worn his clothes."
Summary by our model without syntax annotation: "olivier rousteing has revealed why (he chose) kim and kanye west (up to front) his balmain campaign. french designer (says) the couple are among the most talked-about (people)."
Summary by our model without GAN: "olivier rousteing has revealed why he chose kim and kanye to star in balmain's latest campaign because they represent a family for the new world. the 29-year-old creative director has revealed that he was inspired to feature the couple-who have a 22-month-old daughter north - in the label's spring/summer 2015 men's campaign."

coder generate a "bad" word, the seq2seq could propagate and accumulate errors alongwith the increase of the generated sequence length. The first several words of the generated summaries can be relatively correct, while the quality of sentences deteriorates quickly.

In this paper, we propose a Generative Adversarial Network for Abstractive Text Summarization with Multi-Task constraint (PGAN-ATSMT) to alleviate the aforementioned limitations. Specifically, PGAN-ATSMT jointly trains a generative model G and a discriminative model D via adversarial learning. The generative model G uses the sequence-to-sequence architecture as its backbone, taking the source document as input and generating the summary. We employ reinforcement learning (i.e., policy gradient) to optimize G for a highly rewarded summary. Hence, our model effectively conquers the exposure bias and non-differentiable task metrics issues. The discriminative model D is a language model, and we utilize the output of the language model as the reward to guide the generative model. The generative model G and the discriminative model D are optimized via an adversarial process. The discriminative model D attempts to distinguish the generated summaries by the generative model G from the ground truth summaries, while the generative model G is to maximize the probability of D making a mistake. Consequently, this adversarial process can make the generative model G generate high-quality and plausible abstractive summaries.

Furthermore, we additionally propose two extended regularizations for the generative model G using multi-task learning. First, our LSTM encoder is regularized with the co-training required to perform an additional task of text categorization. Second, our LSTM decoder is also regularized with co-training to provide syntax annotation [30]. This multi-task learning strategy is not prone to maximize the performance of the two auxiliary tasks, but rather to compensate for the missing regularization requirement of the abstractive text summarization task implemented with the seq2seq framework.

Compared with the prior abstractive text summarization methods, the main contributions of this work are as follows:

- We propose *PGAN-ATSMT*, an adversarial framework for abstractive text summarization with multi-task constraint. The adversarial training process of PGAN-ATSMT can eventually adjust the generator to generate plausible and high-quality abstractive summaries.
- PGAN-ATSMT jointly trains the task of abstractive text summarization and two other related tasks: text classification and syntax generation. The auxiliary tasks help to enhance the CNN encoder and the RNN decoder in generating a more satisfied summary for accurately and correctly summarizing the given text.
- We incorporate the retrieved guidance summaries into the encoder-decoder structure, enriching the informativeness and diversity of the generated summaries.
- We conduct comprehensive experiments to evaluate the performance of PGAN-ATSMT model. Experimental results show that PGAN-ATSMT achieves significantly better results than the compared methods on the widely used CNN/Daily Mail and Gigaword datasets.

The rest of this paper is organized as follows. In Section 2, we discuss the related work on abstractive text summarization, generative adversarial networks, and multi-task learning. Section 3 defines the problem. Section 4 briefly introduces the architecture of the PGAN-ATSMT model. In Section 5, we elaborate the sequence to sequence model with multi-task learning. The generative adversarial network for abstractive text summarization is presented in Section 6. In Section 7, we set up the

experiments. The experimental results and analysis are provided in Section 8. Section 9 concludes this manuscript and presents some possible future work.

2. Related work

2.1. Abstractive text summarization

Generally, previous text summarization techniques are categorized as abstractive and extractive. The extractive summarization methods extract salient sentences or phrases from the original articles [43], while the abstractive summarization methods produces new words or phrases, which may rephrase or use words which are not in the raw article [36]. In this manuscript, we mainly work on the abstractive summarization.

In recent years, many efforts have been devoted to developing abstractive text summarization by employing the sequence-to-sequence model [36,37]. For instance, [36] was the first work which employed an encoder-decoder framework with attention mechanism for abstractive summarization. [31] proposed attention encoder-decoder LSTM model to capture the hierarchical document structure and captured the primary words and sentences from the document. [37] introduced a pointer-generator network which allowed both copying the words from the raw document via pointing, and producing words from the vocabulary.

Subsequently, there are some studies attempting to take advantage of both the encoder-decoder LSTM models and the reinforcement learning algorithms for the abstractive text summarization [22,33,41]. For instance, [33] explored both the maximum-likelihood cross-entropy objective and the reinforcement learning objective to alleviate the exposure bias problem [22]. introduced an adversarial process for abstractive text summarization, in which the generator is built as an agent of reinforcement learning.

2.2. Generative adversarial network in text generation

In parallel, previous studies have demonstrated the effectiveness of generative adversarial network (GAN) [11] in various text generation scenarios such as image captioning [5], sequence generation [42] and dialogue generation [20]. GAN tries to train a generator G and a discriminator D jointly, which are optimized with a minimax two-player game. This adversarial process can make the generative model generate plausible and high-quality text sequence. For example, Yu et al. [42] employed a GAN to generate responses, which considered the sequence generation process as the sequential decision making procedure. The generator is treated as a reinforcement learning (i.e., policy gradient) agent, which naturally avoided the differentiation difficulty in discrete data with standard GAN. [5] proposed an adversarial training procedure to leverage unpaired images and sentences in the target domain, which utilized two critic networks to guide the adversarial training procedure. In addition, the policy gradient was applied to update the network of the captioner. Li et al. [20] treated the dialogue generation task as a reinforcement learning problem, jointly training a generative model to generate responses and a discriminative model to distinguish between the human-generated dialogues and the machine-generated dialogues.

2.3. Multi-task learning in natural language processing (NLP)

The goal of multi-task learning is to improve the generalization of the tasks by simultaneously optimizing multiple learning tasks and exploiting the commonalities across tasks [4,10]. For instance, Liu et al. [25] combined the multiple-domain query classification task and the web search task (information retrieval task). The proposed model not only leverages a large scale of data across tasks, but also benefits from the regularization effect which may result in more general sentence representations to improve the new domain tasks. Luong et al. [28] combined the encoder-decoder model with multi-task learning, sharing the parameters of the encoder and decoder across the tasks. The improvement in machine translation is demonstrated. Liu et al. [24] learned several text classification tasks jointly by sharing parameters of RNNs. The performance of the proposed model was evaluated on the two text classification tasks (i.e., subjective classification and sentiment classification). Luan et al. [27] implemented a personalized conversation system by exploiting the multi-task learning to jointly train the neural dialogue systems that uses the dialogue data across speakers.

The main differences between our work and the previous methods can be summarized as the following three aspects: (1) A multi-task learning system is employed to jointly train the abstractive summarization task and two other related tasks: text categorization and syntax annotation. Text categorization co-trained with the summarization model learns a category-specific text encoder and improves the quality of locating salient information of the text. In addition, the LSTM decoder is capable of exploiting word-level syntax to generate high-quality summaries from the language modeling perspective, and thus alleviates the issues of incomplete sentences and duplicated words. (2) The ground truth summaries of similar documents are leveraged to provide a reference point to guide the summarization process of the source documents, inspired by the observation that semantically similar documents tend to share the salient information. (3) A plausibility-promoting generative adversarial network is employed to further refine the abstractive summarization performance, which employs a novel language model based discriminative model to distinguish fluent and novel summaries from implausible and repeated summaries.

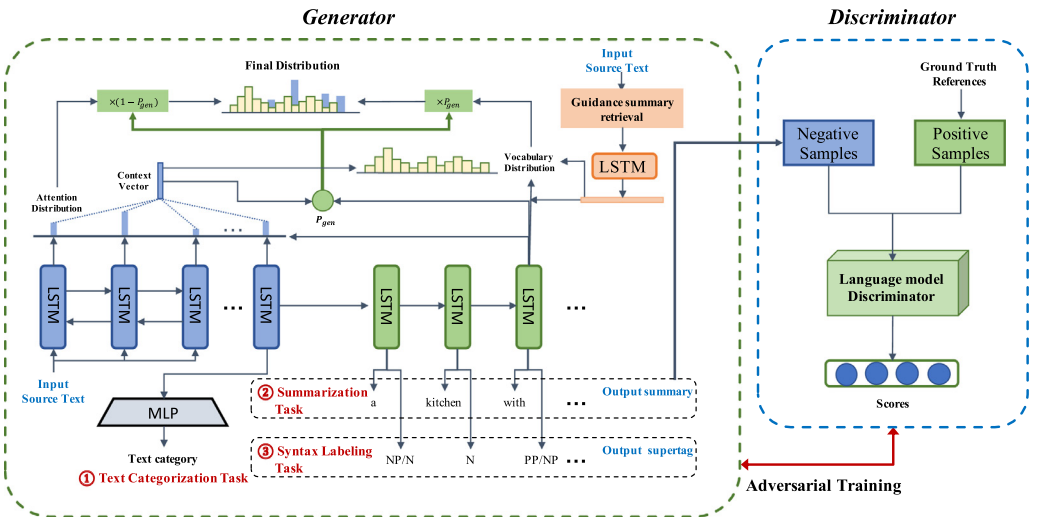

Fig. 1. The overview of the proposed PGAN-ATSMT model.

3. Problem definition

Assume that each source article $X = \{x_1, x_2, \ldots, x_n\}$ has a corresponding reference summary $Y = \{y_1, y_2, \ldots, y_k\}$ and a category label C, where n and k represent the length of the input article and the ground truth summary, respectively. Given the source article X, the goal of abstractive text summarization is to produce a summary $\hat{Y} = \{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_T}\}$, where T denotes the length of $\hat{Y}$. For text categorization, we predict the category label $\hat{C}$ for the input source document. For syntax annotation, a CCG supertag sequence $Z = \{z_1, z_2, \ldots, z_m\}$ is generated for the corresponding summary Y.

4. Architecture of our approach

PGAN-ATSMT model jointly trains a generator G and a discriminator D via an adversarial process. The generative model G uses the sequence-to-sequence architecture as its backbone, which uses an LSTM encoder to compact the document into a hidden representation, and then another LSTM decoder is employed to generate the summary based on the learned document representation. In particular, we additionally propose extended regularizations for the generative model G using multi-task learning. The generator G of PGAN-ATSMT is composed of three key components: (i) a text categorization model that learns the category-aware text representations via an LSTM encoder; (ii) a syntax annotation model that learns a better syntax-aware LSTM decoder; (iii) an abstractive summarization model that shares the encoder and decoder with the text classification task and the syntax annotation task, respectively. In addition, the reinforcement learning technique is employed to optimize G for a highly rewarded summary. The discriminative model D is a language model which is trained to assign scores for the input summaries and the generated summaries. The output of the language model can be used as the reward to guide the generative model. The generative model G and the discriminative model D are optimized with a minimax two-player game.

The framework of PGAN-ATSMT is illustrated in Fig. 1. Next, we will introduce each component of PGAN-ATSMT in details.

5. Multi-task learning for abstractive text summarization

5.1. Shared LSTM encoder

The general idea of the seq2seq framework is to use an LSTM encoder to encode the input text and then feed the representation vector to an LSTM decoder to generate summaries. The abstractive summarizer shares its LSTM encoder with the text categorization task.

In this paper, dependency information is used to capture long distance relationships between two words within a sentence. Dependency information is obtained from the hierarchical tree structure, including dependency features and dependency tags (the relation between current word and its parent node). Formally, for each word x_i from the input article X, we use x_i' to denote its parent word and r_i to represent the dependency relation between x_i and x_i'. The dependency-aware representation of input article X after applying dependency parsing is then represented as:

$$X^p = \{x_1^p, \ldots, x_n^p\} = \{(x_1, x_1', r_1), \ldots, (x_n, x_n', r_n)\} \tag{1}$$

Each word x in the source document is encoded into a low-dimensional embedding $\mathbf{v}_x \in \mathbb{R}^d$ through a word embedding layer, where d denotes the size of word embedding. Similar to the word embedding layer, we also convert each relation r_i

into a relation vector $\mathbf{v}_{r_i} \in \mathbb{R}^d$ by an embedding layer. Then, we employ an LSTM [13] encoder to learn the hidden states of the input article. Mathematically, given the input embeddings at time step t, we compute the hidden state at time step t (i.e., $\mathbf{h}_t$) as:

$$\mathbf{x}_t^p = \tanh(W_1^h \mathbf{v}_{x_t} + W_2^h \mathbf{v}_{x_t'} + W_3^h \mathbf{v}_{r_t}) \tag{2}$$

$$\mathbf{h}_t = \mathbf{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t^p) \tag{3}$$

where W_1^h, W_2^h and W_1^h are parameters to be learned.

After processed by embedding and LSTM layer, the article X can be represented by its hidden states $H = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, where n is the length of article X.

5.1.1. Text categorization task

Text categorization is an auxiliary task to enhance the document representation learning. The goal of the text categorization task is to assign a category label (e.g., "Politics", "Sports") to each input document. Text categorization can be treated as a multi-class classification problem.

For the text categorization task, we use the last item in the hidden vector (i.e., h_n) as the representation of the source text X. The text representation h_n is then fed into a task-specific fully-connected (FC) layer followed by a *softmax* layer to predict the category probability distribution:

$$F^{\text{text}} = \tanh(U_1 \mathbf{h}_n + b_1) \tag{4}$$

$$\hat{C} = softmax(U_2 F^{\text{text}} + b_2), \tag{5}$$

where $\hat{C}$ is prediction probabilities of text categories; U_1, U_2, b_1 and b_2 indicate weight matrices and bias terms, respectively.

The text categorization task is optimized in a supervised manner. To be more specific, given the labeled training data $\{(X_{1:N}, C_{1:N})\}$, we train the text categorization model by minimizing the cross-entropy between the ground truth distribution C and the predicted label distribution $\hat{C}$:

$$J_{\text{ml}}^{\text{text}}(\theta_1) = -\sum_{i=1}^{N}\sum_{j=1}^{J} C_i^j \log(\hat{C}_i^j), \tag{6}$$

where $\hat{C}_i$ is the prediction probabilities of the ith sample, C_i is the ground truth label of the ith sample, J is the number of category classes, N is the number of training samples, θ_1 denotes the parameters related to text categorization model.

5.2. Shared LSTM decoder

The LSTM decoder is a language model, which is conditioned on the output of the encoder. In particular, each token of the target summary is produced by decoding from the distribution over the whole vocabulary. The LSTM decoder is shared by the abstractive summarization task and the syntax annotation task. We use the last item in encoded vector (i.e., h_n) as the initial state of the LSTM decoder. On each decoding step t, the decoder receives the input $\mathbf{u}_t$ (at training, $\mathbf{u}_t$ is the embedding of the previous word of the ground truth summary; while testing, it is the embedding of the previous word emitted by the decoder) and update its hidden state $\mathbf{s}_t$ as:

$$\mathbf{s}_t = \text{LSTM}(\mathbf{s}_{t-1}, [\mathbf{u}_t, \mathbf{c}_t, \mathbf{g}_t]) \tag{7}$$

where $\mathbf{c}_t$ and $\mathbf{g}_t$ are the context vector and guidance vector at time step t. Next, we will give detailed descriptions of $\mathbf{c}_t$ and $\mathbf{g}_t$.

5.2.1. Context vector

The context vector $\mathbf{c}_t$ can be computed as a weighted sum of the hidden states of the encoded input representation H. Formally, we utilize the attention mechanism [2] to compute the attention weights a_t and the context vector $\mathbf{c}_t$ as:

$$\mathbf{c}_t = \sum_{i=1}^{\tilde{n}} \beta_{t,i} \cdot emb_i \tag{8}$$

$$\beta_{t,i} = softmax(f_{t,i}^c) \tag{9}$$

$$f_{t,i}^c = U^{c^T} \tanh(U^{ch} \cdot emb_i + U^{cs} \cdot \mathbf{s}_t + \mathbf{b}^c) \tag{10}$$

where W_h, W_s and b_{attn} are parameters to be learned.

5.2.2. Guidance vector

The guidance vector $\mathbf{g}_t$ can be viewed as the representation of the guidance summary at time step t. According to what we observe, semantically similar articles tend to share the salient objects and events that are often described in their summaries. Thus, the ground truth summaries of similar articles can provide a reference point to guide the summarization process of the input articles. For example, when summarizing an article about the natural disaster, people show a tendency to present the moving path of the disaster and the loss it brings [3]. In this study, we introduce a summary-guided attention model to guide the decoding process, which directly exploits exemplar summaries in the training data as guidance summaries.

Information Retrieval Model We introduce two methods to retrieve similar articles in the training data. One is to leverage the widely-used Information Retrieve (IR) system Lucene,[1] in which the BM25 algorithm [35] is used as the Lucene scoring algorithm to get top-k searching results as candidate guidance summaries. Here, k is an user-defined hyper-parameter. In addition, a sentence embedding based approach is also implemented to do the retrieval based on the semantic representations of sentences. **BM25 Algorithm** Given an query $q = \{w_1, w_2 \ldots, w_n\}$, where w_i represents the token in the query, the BM25 score of a document X is computed as:

$$Score(q, X) = \sum_{i=1}^{n} \frac{IDF(w_i) f(w_i, X)(k_1 + 1)}{f(w_i, X) + k_1(1 - b + b\frac{n}{avg})} \tag{11}$$

where $f(w_i, X)$ is term frequency of w_i in document X, n indicate the number of words in X, avg is the average length of the documents in text collection. k_1 and b are user defined hyper-parameters. Generally, we choose $k_1 \in [1.2, 2.0]$ and $b = 0.75$. $IDF(w_i)$ represents the Inverse Document Frequency weight of word w_i, which is calculated as:

$$IDF(w_i) = \log \frac{N - n(w_i) + 0.5}{n(w_i) + 0.5} \tag{12}$$

where N is the total number of documents in the collection, and $n(w_i)$is the number of documents containing w_i. **Document Embeddings** We employ the method proposed in [1] to learn the document embeddings, which computes the weighted average of the word vectors from the document and then remove the projections of the average vectors on their first principal component. This method gained better performance than other methods on a variety of textual similarity tasks. We utilize the pre-trained word embeddings, e.g., word2vec [29]. For all the article-summary pairs, we build a dictionary *Dict*, whose keys are the document embeddings calculated by the method described proposed in [1], and values are the corresponding summaries. We search top-k similar articles based on cosine similarity over Dict. The corresponding summaries of the selected articles serve as our guidance summaries. **Multiple Guidance Summaries** If the retrieved single summaries is inrelevant, it may lead to misguidance for the decoder. Therefore, we attempt to retrieve multiple guidance summaries instead of one, aiming at improving the stability of our model. The k retrieved guidance summaries are represented as $\{G_1, G_2, \ldots, G_k\}$, where G_i denotes the ith guidance summary. These guidance summaries are concatenated to form a final guidance summary, denoted by $G = [G_1; G_2; \ldots; G_k]$. We employ an LSTM to learn the hidden states H^G of the final summary G.

The guidance vector $\mathbf{g}_t$ can be computed as a weighted sum of the hidden states of the encoded input representation H^G. Formally, we use the attention mechanism [2] to calculate the attention weights $\boldsymbol{\gamma}_t$ and the guidance vector $\mathbf{g}_t$ as

$$\mathbf{g}_t = \sum_{i=1}^{|G|} \gamma_{t,i} \cdot H_i^G \tag{13}$$

$$\beta_{t,i} = softmax(f_{t,i}^g) \tag{14}$$

$$f_{t,i}^g = U^{gT} \tanh(U^{gh} \cdot H_i^G + U^{gs} \cdot \mathbf{s}_t + \mathbf{b}^g) \tag{15}$$

where U^g, U^{gh}, U^{gs} and $\mathbf{b}^g$ are learnable parameters.

5.2.3. Introduction of CCG supertag annotation

Combinatory Category Grammar (CCG) [38] utilizes a set of lexical categories to represent constituents, which provides a connection between semantics and syntax of natural language. In particular, a fixed finite set of CCG categories is reported in Table 3. The basic lexical categories can be utilized to produce an infinite set $\mathbb{C}$ of functional categories by using the recursive definition: (i) $N, NP, S, PP \in \mathbb{C}$; (2) $A/B, A\backslash B \in \mathbb{C}$ if $A, B \in \mathbb{C}$.

CCG supertag annotation [9] is the task of assigning lexical categories to each word in a piece of text. Formally, CCG supertag annotation can be formulated by $P(Z|X)$, where $X = \{x_1, \ldots, x_n\}$ indicates the n words in a document, and $Z = \{z_1, \ldots, z_n\}$ represents the corresponding lexical categories. Please note that the size of the document and the CCG supertag sequence is the same. We provide two examples of documents and the corresponding CCG supertags in Table 4.

[1] https://lucene.apache.org/.

Table 3
The basic lexical categories used in combinatory category grammar.

CCG category	Description
N	noun
NP	noun phrase
PP	prepositional phrase
S	sentence

Table 4
Examples of sentences and corresponding CCG supertags generated by our model.

Captions:	a	kitchen	with	two	pots	sitting	on	a	stove	
CCG:	NP/N	N	PP/NP	NP/N	N	(S[ng]\NP)/(S\NP)\(S\NP)	(S\NP)\(S\NP)/NP	NP/N	N	
Captions:	a	suitcase	filled	with	lots	of	items	on	a	bed
CCG:	NP/N	N	(S[pss]\NP)/PP	PP/NP	NP/PP	PP/NP	NP	(S\NP)\(S\NP)/NP	NP/N	N

Table 5
Key features and hyperparameters used in our model.

Hyperparameters	Values
Learning rate	0.0001
Size of each word vector	100
Number of training epochs	15
Mini-batch training size	16
Size of beam search	5
Number of hidden states for LSTM	256
Number of feature maps for CNN	200
Width of convolution filters	2
Weight decay value of L_2 regularization	0.0001
Dropout rate	0.2
Function used to initiate recurrent parameters	Orthogonal matrices
Function used to initiate weights	$\mathcal{N}(0, 0.01)$
Function used to initiate biases	Zero vector
Optimization algorithm	Adam

5.2.4. Syntax annotation and abstractive summarization tasks

We then concatenate the context vector $\mathbf{c}_t$, the guidance vector $\mathbf{g}_t$ and the decoder hidden state $\mathbf{s}_t$ at time step t and feed it to a linear function to produce the hidden vector of the decoder:

$$O_t = V[\mathbf{s}_t, \mathbf{c}_t, \mathbf{g}_t] + \mathbf{b} \tag{16}$$

The generation probabilities of the t-th word and CCG supertag are computed as:

$$P_t^{\text{sum}} = P^{\text{sum}}(y_t|Y_{1:t-1}; X) = \text{softmax}(U^{\text{sum}}O_t + b^{\text{sum}}) \tag{17}$$

$$P_t^{\text{syntax}} = P^{\text{syntax}}(z_t|Y_{1:t-1}; X) = \text{softmax}(U^{\text{syntax}}O_t + b^{\text{syntax}}) \tag{18}$$

where the U^{sum}, U^{syntax}, b^{sum}, b^{syntax} are parameters to be learned. The superscripts *syntax* and *sum* denote the parameters related to supertag annotation and abstractive summarization, respectively. $y_{1:t-1}$ denotes the previously generated tokens. Note that P_t^{sum} denotes the word distribution over the whole vocabulary at time step t.

Nevertheless, the standard text generation model may suffer from the out-of-vocabulary problem and produce many "UNK" tokens in the summary. To alleviate this limitation, copy mechanism is widely adopted in recent abstractive summarization systems [14,15,37]. Similar to the work [37], in this study the generation probability $p_{\text{gen}} \in [0, 1]$ at time step t is computed from the decoder state s_t, context vector c_t, and decoder input u_t:

$$P_{\text{gen}} = \sigma(V_c^T c_t + V_s^T s_t + V_u^T u_t + b_{gen}) \tag{19}$$

where vectors V_c, V_s, V_u and scalar b_{gen} are learnable parameters.

For each step t, given a candidate token w_j (j denotes the index of the vocabulary), if w_j is out-of-vocabulary token, then $p_t^w(w_j) = 0$, if it does not appear in the source text, then $a_{t,j} = 0$.

$$\tilde{P}_t^{\text{sum}}(w_j) = P_{\text{gen}} * P_t^{\text{sum}}(w_j) + (1 - P_{\text{gen}}) * \sum a_{t,j} \tag{20}$$

For the syntax labeling and summarization generation subtasks, we employ the minimum negative log-likelihood estimation: Specifically, the objective is the sum of negative log likelihood of the target word/supertag at each decoding step.

$$J_{\text{ML}}^{\text{sum}}(\theta_2) = -\sum_t^T \log(\tilde{P}_t^{\text{sum}}) \tag{21}$$

$$J_{\mathrm{ML}}^{\mathrm{syntax}}(\theta_3) = -\sum_{t}^{T} \log(P_t^{\mathrm{syntax}}) \tag{22}$$

where T represents the length of the output sequence during the decoding phase.

5.2.5. Joint training

In order to improve the shared LSTM encoder and LSTM decoder, we optimize these three related tasks simultaneously. The joint multi-task objective function is minimized by:

$$J_{\mathrm{ML}}(\Theta) = \lambda_1 J_{\mathrm{ML}}^{\mathrm{text}} + \lambda_2 J_{\mathrm{ML}}^{\mathrm{sum}} + \lambda_3 J_{\mathrm{ML}}^{\mathrm{synax}} \tag{23}$$

where Θ denotes the collective parameters of the model. λ_1, λ_2 and λ_3 are hyper-parameters that determine the weights of the three objectives. Here, we set $\lambda_1 = \lambda_2 = 0.45$, and $\lambda_3 = 0.1$. The parameters for multitask learning are determined by performing the grid search on a validation set.

5.2.6. Policy gradient reinforcement learning for summary generation

Nevertheless, the maximum likelihood estimation (MLE) based methods have several limitations. First, the automatic metrics are different from the training loss. For instance, in abstractive text summarization systems, the encoder-decoder framework is trained with the cross-entropy loss. Nevertheless, at test time, the model is typically evaluated with non-differentiable and discrete metrics such as BLEU [32] and ROUGE [21]. Second, at each time step in the training phase, the decoder usually takes as input the previous ground-truth word. However, when generating summaries during testing phase, the input of the decoder at next time step is the previous word produced by the decoder. This exposure bias problem [34] results in error accumulation while testing. Once the model produces a "bad" word, the error will accumulate alongwith the increase of the sequence length.

To alleviate the above limitations when decoding summaries, we also optimize directly for ROUGE-1 by employing the policy gradient to maximize the expected rewards:

$$J_{\mathrm{RL}}^{\mathrm{sum}} = (r(\hat{y}) - r(y^s)) \sum_{t}^{N_x} \log \tilde{P}_t^{\mathrm{sum}}(y_t^s | Y_{1:t-1}^s; X) \tag{24}$$

where N_x is the length of the article X, $r(\hat{y})$ represents the reward of the generated sequence $\hat{y}$ by greedy decoding, and $r(y^s)$ represents the reward of sequence y^s that is generated by sampling among the vocabulary per step.

After pre-training PGAN-ATSMT by optimizing the joint multi-task objective (refer to Eq. (23)), we change PGAN-ATSMT to further optimize a mixed training objective, which integrates the multi-task objective $J_{\mathrm{ml}}(\theta)$ with the reinforcement learning objective $J_{\mathrm{RL}}^{\mathrm{sum}}(\theta)$:

$$J_{\mathrm{mixed}}(\Theta) = \beta J_{\mathrm{ML}}(\Theta) + (1-\beta) J_{\mathrm{RL}}^{\mathrm{sum}}(\Theta) \tag{25}$$

where β is a hyper-parameter, and we set β=0.1, Θ denotes the set of parameters of the encoder-decoder framework.

6. Plausibility-promoting generative adversarial network

We propose a plausibility-promoting generative adversarial network (PP-GAN) to refine the performance of text summarization. The basic structure of PP-GAN is similar to that of the standard GAN [11], which consists of consists of a generative model G and a discriminative model D, which compete in a minimax game. The discriminative model tries to distinguish the real summaries in the training dataset from the generated summaries by G, while the generative model G tries to fool the discriminative model by generating plausible and human-like summaries. Formally, D and G play the following game on $L(D, G)$:

$$\min_G \max_D L(D,G) \quad = \mathbb{E}_{X\sim P_{true}(X)}[\log D(X)] + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{26}$$

Here, x is ground truth data from the training set, z is the noise variable randomly sampled from the normal distribution.

6.1. Discriminative model D

Typically, most previous GAN based text generation methods employ a binary text classifier as the discriminative model D, aiming to distinguish the input text as generated by the generative model G or originally written by humans. However, using a binary classifier as the discriminative model usually struggles to achieve satisfactory results. A binary classifier may obtain high accuracy, making the generated summaries receive the reward around zero since the discriminative model is able to predict the generated summaries with high confidence. Instead of applying a binary classifier as the discriminative model D, we employ a language model to implement D and utilize the output of the language model as the reward to guide the generative model.

In the adversarial learning, we utilize the discriminator as a reward function to guide the generative model. The parameters of the discriminator D is optimized by maximizing the reward of ground-truth summary while minimizing the

reward of the summary generated by G. In this way, the low-quality summaries generated by G can be recognized by D easily and receive low reward, guiding the generative model G to produce the summaries that looks like the ground-truth (human-written) summaries. The parameters of discriminator D and generator G are optimized iteratively and alternately. Once acquiring more high-quality and plausible summaries generated by the generative model, we can re-train the discriminative model as:

$$\min_{\phi} -\mathbf{E}_{Y\sim p_{data}}[logD_{\phi}(Y)] - \mathbf{E}_{Y\sim G_{\Theta}}[log(1-D_{\phi}(Y))] \tag{27}$$

where ϕ and Θ represent the parameter sets of discriminator D and generator G.

6.2. Generative model G

When the discriminative model D is optimized and fixed, we are prepared to update the generative model G. The objective function of the generative G is defined by Eq. (25). According to policy gradient theorem [39], we calculate the gradient of J_{mixed} w.r.t. the parameter set Θ as:

$$\begin{aligned}\nabla_{\Theta} J_{\text{mixed}} &= \frac{1}{T}\sum_{t=1}^{T}\sum_{\hat{y}_t} R_t \cdot \nabla_{\Theta}(G_{\Theta}(\hat{y}_t|\hat{Y}_{1:t-1}, X)) \\ &= \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}_{\hat{y}_t\in G_{\Theta}}[R_t \nabla_{\Theta} \log \tilde{P}_t^{\text{sum}}(\hat{y}_t|\hat{Y}_{1:t-1}, X)]\end{aligned} \tag{28}$$

where T represents the length of the generated sequence, $\hat{Y}_{1:t}$ is the partial summary generated by G up to time step t, R_t represents the total reward starting from step t.

Inspired by Xu et al. [40], we combine both word-level and sentence-level rewards to form the total reward R_t, which evaluates the quality of the generated summary. Concretely, given a generated summary $\hat{Y}_{1:T}$, the sentence-level reward is the averaged reward of each word $\hat{y}_t$, which is defined as:

$$R(\hat{Y}_{1:T}) = -\frac{1}{T}\sum_{t=1}^{T}\log D_{\phi}(\hat{y}_t|\hat{Y}_{1:t-1}) \tag{29}$$

Different words in the summary should have different rewards. For example, salient topic words are much more important than the stop words. Thus, we also calculate the word-level reward for each token $\hat{y}_t$:

$$R(\hat{y}_t) = -\log D_{\phi}(\hat{y}_t|\hat{Y}_{1:t-1}) \tag{30}$$

Finally, the total reward R_t starting from step t is computed by:

$$R_t = \sum_{k=t}^{T} R(\hat{Y}_{1:T})R(\hat{y}_k) \tag{31}$$

7. Experimental setup

7.1. Datasets description

Extensive experiments are conducted on two widely used real-life datasets. The detailed properties of the datasets are described as follows:

- **CNN/Daily Mail Corpus** We first evaluate our model on the CNN/Daily Mail Corpus [16], which is widely used in abstractive text summarization. The dataset comprises news stories in CNN/Daily Mail websites paired with multi-sentences human-generated summaries. Totally, it consists of 287,226 training instances, 13,368 validation instances and 11,490 test instances. There are 781 tokens on average of articles and 56 tokens on average of summaries.
- **Gigaword Corpus** The Gigaword corpus is originally introduced by Graff et al. [12]. Following [31], we utilized the public available scripts to preprocess the data.[2] Totally, there are about 3.8M training instances, 400K validation and test instances.

In all experiments, data preprocessing is performed. Each text is tokenized with a widely used natural language processing toolkit NLTK.[3] We build independent vocabularies for articles and summaries by keeping the top 20,000 words with the highest frequency. The rest words that are not included in the vocabularies are displaced by the "*UNK*" token.

For text categorization, we explore the source webpage of each news article, where a specific category is provided for each article. This dataset covers 11 categories: Sports, Showbiz, Politics, Opinion, Tech, Travel, Health, Crime, Justice, Living and Business.

[2] Code is available at https://github.com/kyunghyuncho/dl4mt-material.

[3] http://www.nltk.org.

For syntax annotation, each summary is annotated with CCG supertags,[4] in which each word has a corresponding dependency label of supertags.

7.2. Implementation details

Following the settings of [37], we use the non-anonymized version and truncate the input articles/target summaries to a maximum length of 400/100 words. We use the 100-dimensional word2vec [29] embeddings trained on the 2014 dumps of English Wikipedia to initialize the word embeddings for both datasets (i.e., $d = 100$). For both datasets, we initialize the recurrent parameter matrices as orthogonal matrices, and the other parameters are initialized with the normal distribution $\mathcal{N}(0, 0.01)$. These word embeddings are fine-tuned when training the model. Our LSTM encoder and LSTM decoder have hidden state size of 256. For the convolutional layer of discriminative model D, we set both the number of feature maps of CNN to 200. The width of the convolution filters is set to be 2.

We first pretrain ML model for summarization with a learning rate of 0.15 [37]. Then switch to *PGAN-ATSMT* training using the Adam optimizer [19], with mini-batch size of 16 and a learning rate of 0.0001. We use a beam search with beam size of 5 during decoding. Dropout (with the dropout rate of 0.2) and L_2 regularization (with the weight decay value of 0.0001) are used to avoid overfitting.

7.3. Baseline methods

In this paper, we compare the proposed model with several state-of-the-art baseline methods:

- **ABS and ABS+** Encoder-decoder RNNs with attention networks are proposed for abstractive summarization [31].
- **RAS-LSTM and RAS-Elman** Two seq2seq architectures with attention mechanism are employed to abstractive text summarization [8]. RAS-LSTM uses LSTM network as decoder while RAS-Elman adopts Elman RNN as decoder.
- **PGC** This is the pointer-generator coverage network that is originally proposed in [37], which copies words from the document through pointing and retains the ability to generate novel words via the generator.
- **DeepRL** The deep reinforced model (ML+RL version) is proposed by Paulus et al. [33]. A new loss function is proposed, which combines the MLE objective function and the reinforcement learning objective function to alleviate the exposure bias problem.
- **GANsum** The generative adversarial network is utilized to develop abstractive text summarization [23].
- **ConvKG** A convolutional recurrent framework is adopted by Chen et al. [6]. In addition, the keyphrases are leveraged to guide the decoding process to produce more well-formed and abstractive summaries.
- **MATS** This method presents a multi-task learning framework, which explores two auxiliary tasks to enhance the abstractive text summarization task with reinforcement learning [26].

8. Experimental results

In the experiments, we evaluate the proposed PGAN-ATSMT model from both quantitative and qualitative perspectives.

8.1. Automatic evaluation results

Following the same evaluation as in previous work [31], we evaluate PGAN-ATSMT using three widely used automatic evaluation metrics including ROUGE-1, ROUGE-2 and ROUGE-L. ROUGE-N [21] is widely adopted in evaluating the summarization tasks, which estimates the consistency between the n-gram occurrences in the reference summaries and the generated summaries. ROUGE-L compares the longest common sequence between the reference summaries and the generated summaries. We also evaluate the effectiveness of PGAN-ATSMT with perplexity [18] that is widely used to measure how well the model predicts a sequence as a language model. A lower perplexity score indicates the better performance of PGAN-ATSMT.

We report the ROUGE and perplexity scores of PGAN-ATSMT model and the compared methods in Tables 6–7. From the results, we can make the following observations:

- The proposed PGAN-ATSMT consistently and substantially outperforms the compared methods by a noticeable margin on both datasets. This verifies the effectiveness of our model for abstractive text summarization.
- The models which incorporate the copying strategy into the encoder-decoder framework (e.g., PGC) perform better than the standard generative models (e.g., ABS, ABS+, RAS-LSTM, RAS-Elman). This may be because that the pointer-generator network can handle the out-of-vocabulary words and retain the capability of generating new words.
- DeepRL, GANsum and MATS, which combine the strengths of both supervised deep learning and reinforcement learning, achieve better performance than the other baseline methods, because they utilize reinforcement learning to alleviate the exposure bias problem and optimize directly the evaluation metrics.

[4] https://github.com/uwnlp/EasySRL.

Table 6
Quantitative evaluation results (ROUGE and perplexity scores) for CNN/Daily Mail test data. We use pyrouge, a Python wrapper of the ROUGE script with parameter "-c 95 -2 -1 -U -r 1000 -n 2 -w 1.2 -a" to compute the ROUGE scores.

Methods	ROUGE-1	ROUGE-2	ROUGE-L	Perplexity
ABS	35.46	13.30	32.65	17.49
ABS+	35.63	13.75	33.01	17.04
RAS-LSTM	37.46	15.11	34.45	15.24
RAS-Elman	38.25	16.28	35.43	14.52
PGC	39.53	17.28	36.38	12.36
DeepRL	39.87	15.82	36.90	12.43
GANsum	39.92	17.65	36.71	11.75
ConvKG	40.01	17.46	36.63	13.19
MATS	40.74	18.14	37.15	12.72
PGAN-ATSMT	**42.15**	**19.98**	**38.94**	**10.21**

Table 7
Quantitative evaluation results (ROUGE and perplexity scores) for Gigaword test data. We use pyrouge, a Python wrapper of the ROUGE script with parameter "-c 95 -2 -1 -U -r 1000 -n 2 -w 1.2 -a" to compute the ROUGE scores.

Methods	ROUGE-1	ROUGE-2	ROUGE-L	Perplexity
ABS	29.55	11.32	26.42	18.92
ABS+	29.78	11.89	26.97	18.46
RAS-LSTM	32.55	14.70	30.03	17.24
RAS-Elman	33.78	15.97	31.15	16.49
PGC	33.44	16.09	31.43	14.90
DeepRL	35.16	16.75	31.68	15.34
GANsum	35.04	16.55	31.96	14.37
ConvKG	35.16	16.72	32.16	15.42
MATS	35.56	16.97	32.94	14.63
PGAN-ATSMT	**37.83**	**19.15**	**34.72**	**12.78**

- Even the better-performing model generates summaries with low ROUGE-2 results, indicating the low rate of 2-gram overlap between the generated summaries and the ground-truth summaries. This suggests that a keyphrase detector should be developed to identify the salient information.

8.2. Human evaluation

The automatic metrics ROUGE-N and ROUGE-L have wide adoption in evaluating abstractive summarization systems. However, how well the existing automatic metrics are matched with the human judgement is still controversial. In this paper, we also perform the human evaluation to estimate the *relevance* and *readability* of the summaries that are generated by PGAN-ATSMT and the compared methods. Specifically, we choose 200 cases from the test data. Similar to [7], three NLP researchers are asked to give each generated summary an integer score of 1 (bad), 2 (poor), 3 (not bad), 4 (satisfactory), 5 (good) for *relevance* and *fluency*, separately. For the *relevance* part, we check if the generated summary contains the salient information of the article; while for the fluency part, we check if the generated summary is grammatically correct. The human evaluation results are reported in Table 8. The proposed PGAN-ATSMT model significantly outperforms the compared approaches by a large margin on both datasets. Specifically, PGAN-ATSMT improves 4.4% and 4.6% on the *relevance* and *fluency* scores over the best results of baseline methods on the CNN/Daily Mail test data. However, the summaries generated by the models were not as relevant as the reference summaries would likely be. The overarching tendency of the models is still to copy segments of the source document. It is still challenging to produce informative, fluent, and abstractive summaries.

8.3. Ablation study

To estimate the impact of each part on the performance of PGAN-ATSMT, we perform the ablation test of PGAN-ATSMT in terms of removing text categorization (denoted as w/o text), removing syntax generation (denoted as w/o syntax), removing generative adversarial network framework (denoted as w/o GAN), removing dependency information in encoding (w/o dependency), removing guidance vector in decoding (w/o guidance), and replacing the reinforcement learning with Gumbel-Softmax [17] (denoted as w/o RL), respectively. In addition, we also investigate the impact of language model based discriminator D by replacing the language model with a CNN based binary classifier (denoted as w/o LM-D). The ablation results are summarized in Tables 9 and 10.

Table 8
Human evaluation results for CNN/Daily Mail and Gigaword datasets.

Methods	CNN/Daily Mail		Gigaword	
	Relevance	Fluency	Relevance	Fluency
ABS	2.75	2.94	2.58	2.77
ABS+	2.79	2.92	2.61	2.83
RAS-LSTM	2.86	2.97	2.65	2.89
RAS-Elman	2.91	3.04	2.69	2.90
PGC	3.02	3.09	2.76	2.98
DeepRL	3.04	3.03	2.79	2.95
GANsum	3.15	3.19	2.91	3.07
ConvKG	3.08	3.15	2.96	3.10
MATS	3.11	3.23	2.94	3.16
PGAN-ATSMT	**3.29**	**3.38**	**3.12**	**3.27**

Table 9
Ablation test results for CNN/Daily Mail dataset.

Methods	ROUGE-1	ROUGE-2	ROUGE-L	Perplexity
PGAN-ATSMT	**42.15**	**19.98**	**38.94**	**10.21**
w/o text	40.98	19.13	37.95	11.18
w/o syntax	41.77	19.54	38.43	11.56
w/o RL	41.12	18.79	37.67	11.32
w/o GAN	40.84	18.97	38.06	12.15
w/o dependency	41.75	19.54	38.59	10.47
w/o guidance	41.42	19.26	38.45	10.63
w/o LM-*D*	41.36	19.33	38.34	11.97

Table 10
Ablation test results for Gigaword dataset.

Methods	ROUGE-1	ROUGE-2	ROUGE-L	Perplexity
PGAN-ATSMT	**37.83**	**19.15**	**34.72**	**12.78**
w/o text	36.83	18.15	33.59	13.52
w/o syntax	37.42	18.84	34.32	13.95
w/o RL	36.35	17.82	33.46	14.10
w/o GAN	36.72	18.25	33.76	13.84
w/o dependency	37.47	18.93	34.46	13.12
w/o guidance	37.25	18.64	34.22	13.43
w/o LM-*D*	37.26	18.96	34.35	14.05

Generally, all the factors contribute great improvement to PGAN-ATSMT. From Tables 9 and 10, we have several key observations:

- The ROUGE scores decrease sharply when removing the generative adversarial network framework. This is within our expectation since the RL reward signal coming from the discriminative model of GAN guides the model to enjoy considerable success in generating plausible summaries.
- Discarding reinforcement learning leads the adversarial learning inefficient. This is because the generative model G does not benefit from the reward of the discriminative model D when replacing the RL algorithm with Gumbel-Softmax. Therefore, removing RL algorithm has biggest impact on the performance of PGAN-ATSMT.
- The text categorization task also contributes to the effectiveness of PGAN-ATSMT. This verifies that the text categorization helps to learn better category-aware representations and locate salient information.
- The syntax annotation task also makes a contribution to PGAN-ATSMT. However, the improvement of integrating syntax annotation is relatively limited. This may be because that the issue of the incomplete sentence has little effect on the evaluation metrics of summarization.

8.4. Case study

To estimate PGAN-ATSMT qualitatively, we report some example summaries that are generated by PGAN-ATSMT and the compared methods. Due to limited space, we choose one generated summary by DeepRL, GANsum and PGAN-ATSMT from test data for comparison. We report the generated summaries in Table 11. We have several key observations from Table 11:

Table 11
Example summaries. The words in red indicate the incorrect or redundant phrases, while the words in blue represent the words that we need to insert to make the whole summary fluent and grammatically correct.

Input: "Article (truncated): the owners of two dogs that got loose and mauled a jogger to death in michigan pleaded no contest on friday to a reduced charge. sebastiano quagliata and his wife, valbona lucaj, pleaded no contest to owning a dangerous dog causing death in the fatal mauling last summer of craig sytsma of livonia, the flint journal reported. ' i'm so sorry', lucaj told sytsma's family. sytsma was attacked by two cane corsos last july in metamora township, 45 miles northwest of detroit. sebastiano quagliata (pictured) and his wife, valbona lucaj, pleaded no contest to owning a dangerous dog causing death in the fatal mauling of craig sytsma of livonia, michigan, and agreed to 15 years in jail. lucaj (left), a native of albania, and quagliata, of italy, could be deported after serving their sentences. sytsma was attacked by two cane corsos last july in metamora township, 45 miles northwest of detroit. the father-of-three was bitten some eight times by two hundred-pound cane corsos while jogging in metamora township in july, and was 'screaming and begging' for help in his final minutes. lapeer county prosecutor tim turkelson predicts a sentence of at least 19 months in prison when the two return to court on june 8. (... ...) the attack was the third since 2012 involving canines from lucaj and quagliata 's property, officials said. ' animal control should have done something, ' said april smith, who was bitten in may 2012 while walking her two dogs. cane corsos are italian dogs that are ' not to be trifled with ', according to the american kennel club. the akc report about the breed said : ' topping out at nearly 28 in. at the shoulder and often weighing more than 100 pounds, with a large head, alert expression, and muscular frame, cane corsi are at a glance intimidating creatures. ' ' bred as guard dogs for centuries, their imposing appearance is their first line of defense against intruders. ' "
Reference summary: "father-of-three craig sytsma was mauled to death in michigan last year. dog owners sebastiano quagliata and wife valbona lucaj agreed to plea deal friday to around 15 years in jail, though judge could add six months. Sytsma was jogging in july 2014 when the two cane corsos attacked him. He was bit10 almost ten times and was 'screaming and begging' for help." "
Summary by DeepRL: "sebastiano quagliata and his wife, valbona lucaj, pleaded no contest to dog death death, the flint journal reported. sebastiano quagliata (pictured) and his wife, valbona lucaj, pleaded no contest to craig sytsma of livonia livonia linonia."
Summary by GANsum: "sytsma was attacked by two hundred-pound cane corsos. lucaj, a native of albania, and his wife, valbona lucaj, pleaded no contest to owning a dangerous dog causing death in fatal mauling last summer."
Summary by PGAN-ATSMT: "sebastiano quagliata and his wife, valbona lucaj, pleaded no contest to owning a dangerous dog causing death in the fatal mauling of craig sytsma of livonia, and agreed to 15 years in jail. sytsma was bitten eight times by two hundred-pound cane corsos while jogging in metamora township in july, and was screaming and begging for help in his final minutes."
Summary by PGAN-ATSMT without syntax: "sytsma was attacked by two [cane corsos] last july, metamora township. lucaj, a native of albania, and his wife, valbona lucaj, pleaded no contest to [owning a dangerous] dog [causing] death in fatal mauling last summer."
Summary by PGAN-ATSMT without text categorization: "lucaj, a native of albania, and quagliata, of italy, could be deported after serving their sentences. sebastiano quagliata pleaded no contest to owning a dog, the flint journal reported."
Summary by PGAN-ATSMT without GAN: "sebastiano quagliata and his wife, valbona lucaj, pleaded no contest to owning a dangerous dog. sytsma was attacked by two cane corsos last july in metamora township, 45 miles northwest of detroit. lucaj was bitten some eight times by two hundred-pound cane corsos while jogging in metamora township in july, and was 'screaming and begging' for help in his final minutes."

- Although DeepRL achieves high ROUGE scores, it generates the summary that is repeated and less readable. The most common readability problem is the presence of the repeated or grammatically incorrect phrases in the summary. This suggests that optimizing for the single evaluation metric (e.g., ROUGE) with RL can be detrimental to the model quality.
- Existing methods that do not consider the text categories tend to generate trivial and generic summaries that easily miss or under-represent salient aspects of the original document (see the summary generated by PGAN-ATSMT without text categorization), confirming the necessity of the text categorization auxiliary task.
- The syntactic information plays a crucial role in sentence generation. Enforcing syntactic conformance addresses issues like incomplete sentences and duplicated words. As shown in Table 11, the model which is unaware of the text syntax could generate a broken summary for the given document. Yet, an improved system whose summarization component has been co-trained with syntax annotation task could generate a more satisfied sentence for accurately and correctly condensing the raw document.
- As shown in Table 11, GAN can further refine the summarization performance and generate more plausible, high-quality and human-like abstractive summaries by adding/deleting information and rephrasing the generated words.
- The overarching tendency of PGAN-ATSMT is still to copy substrings of the input document and rearrange the phrases into a summary. A new strategy should be developed generate more new concepts and phrases that retain the salient information of the input document.

8.5. Learning curve

To better understand the training process of PGAN-ATSMT, we illustrate the learning curves of PGAN-ATSMT as demonstrated in Fig. 2. Due to the space limitation, we only show the learning curves with respect to ROUGE-L for both CNN/Daily and Gigaword datasets. ROUGE-1 and ROUGE-2 exhibit the similar trend. As shown in Fig. 2, during pre-training, PGAN-ATSMT converges after about 8 epochs for CNN/Daily and 13 epochs for Gigaword. The ROUGE-L scores are further improved on both datasets by applying the GAN, verifying that the generative model becomes better with the effective feedback (reward) from the discriminative model.

8.6. Computational cost

Since it is difficult to analyze the time complexity of deep learning models theoretically, we investigate the computational cost of the models on CNN/Daily Mail dataset, instead of analyzing the time complexity. We train the proposed PGAN-ATSMT model on a NVIDIA Tesla P100 GPU. At the training phase, PGAN-ATSMT spends about 8.5 hours per epoch for CNN/Daily

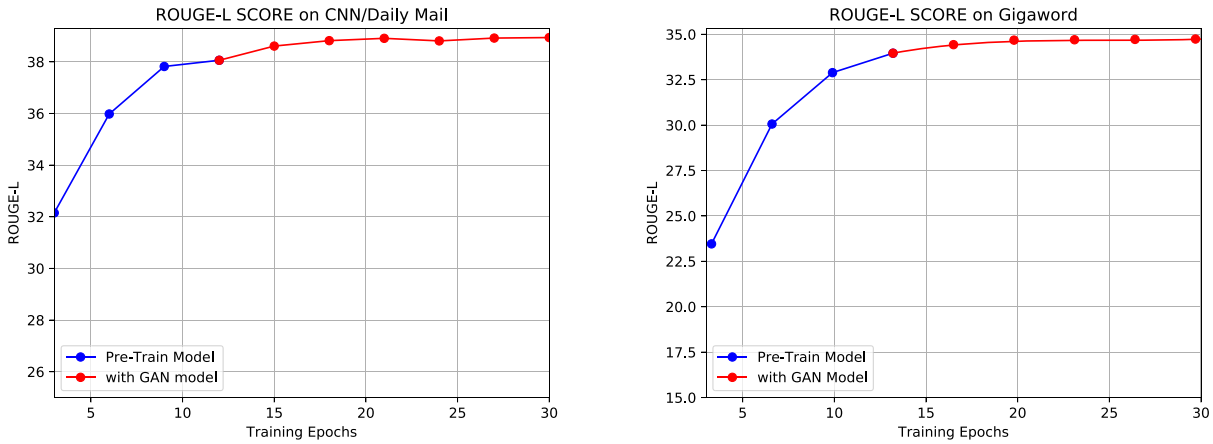

Fig. 2. Learning curves of our model in terms of Rouge-L on CNN/Daily and Gigaword datasets.

dataset. As presented in [31], most compared baseline models take approximately 4–8 hours every epoch on an average. Almost all the compared models converge within 15 epochs by applying the early stopping strategy. At the testing phase, the generation of summaries is fairly fast with a throughput of approximately 28 summaries every second with the batch size of 1.

9. Conclusions and future work

In this paper, we described a novel Generative Adversarial Network for Abstractive Text Summarization with Multi-Task constraint (PGAN-ATSMT). Our model jointly trains a generator G and a discriminator D via adversarial learning. The generative model G uses the sequence-to-sequence architecture as its backbone, taking the source document as input and generate the summary. Instead of applying a binary classifier as the discriminative model D, we employ a language model to implement D and utilize the output of the language model as the reward to guide the generative model. The generative model G and the discriminative model D were optimized with a minimax two-player game. Thus, this adversarial process could eventually adjust G to generate plausible and high-quality abstractive summaries. In addition, the generative model G of PGAN-ATSMT also enjoyed significant benefit from two additional auxiliary tasks: text categorization and syntax annotation. The auxiliary tasks help to improve the quality of locating salient information of a document and generate high-quality summaries from language modeling perspective, which alleviates the issues of incomplete sentences and duplicated words. To estimate the effectiveness of PGAN-ATSMT, we conducted comprehensive experiments on two widely used abstractive summarization datasets. Experimental results demonstrated that PGAN-ATSMT achieved higher ROUGE score and human evaluation than several strong baseline models. Furthermore, the human evaluation also verified that the proposed model could generate summaries with better readability.

In the future, we may devote our effort to explore automatic evaluation metrics that may better match the human judgments. In addition, we also plan to incorporate external commonsense knowledge from the knowledge base or WordNet into the deep neural networks, which may help generate more precise and comprehensive text summaries.

Declaration of Competing Interest

There is no conflict of interest.

CRediT authorship contribution statement

Min Yang: Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Xintong Wang:** Data curation, Software, Writing - original draft. **Yao Lu:** Data curation, Software, Writing - original draft. **Jianming Lv:** Methodology, Supervision. **Ying Shen:** Software, Validation. **Chengming Li:** Writing - review & editing, Supervision.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (No. 61906185), the Natural Science Foundation of Guangdong Province of China (No. 2019A1515011705), the SIAT Innovation Program for Excellent Young Researchers (Grant No. Y8G027).

References

[1] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings (2016).
[2] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations, 2014.
[3] Z. Cao, W. Li, S. Li, F. Wei, Improving multi-document summarization via text classification., in: AAAI, 2017, pp. 3053–3059.
[4] R. Caruana, Multitask Learning, in: Learning to Learn, Springer, 1998, pp. 95–133.
[5] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, M. Sun, Show, adapt and tell: adversarial training of cross-domain image captioner, in: The IEEE International Conference on Computer Vision (ICCV), 2, 2017, pp. 521–530.
[6] X. Chen, J. Li, H. Wang, Keyphrase guided beam search for neural abstractive text summarization, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–9.
[7] J. Cheng, M. Lapata, Neural summarization by extracting sentences and words, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 484–494.
[8] S. Chopra, M. Auli, A.M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93–98.
[9] S. Clark, Supertagging for combinatory categorial grammar, in: Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 6), 2002, pp. 19–24.
[10] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167.
[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[12] D. Graff, J. Kong, K. Chen, K. Maeda, English gigaword, Linguist. Data Consortium, Philadelphia 4 (1) (2003) 34.
[13] A. Graves, S. Fernández, J. Schmidhuber, Bidirectional lstm networks for improved phoneme classification and recognition, in: International Conference on Artificial Neural Networks, Springer, 2005, pp. 799–804.
[14] J. Gu, Z. Lu, H. Li, V.O. Li, Incorporating copying mechanism in sequence-to-sequence learning, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 1, 2016, pp. 1631–1640.
[15] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, Y. Bengio, Pointing the unknown words, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1, 2016, pp. 140–149.
[16] K.M. Hermann, T. Kočiskỳ, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, MIT Press, 2015, pp. 1693–1701.
[17] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv:1611.01144(2016).
[18] F. Jelinek, R.L. Mercer, L.R. Bahl, J.K. Baker, Perplexitya measure of the difficulty of speech recognition tasks, The Journal of the Acoustical Society of America 62 (S1) (1977). S63–S63
[19] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv:1412.6980(2014).
[20] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, arXiv:1701.06547(2017).
[21] C.-Y. Lin, Rouge: a package for automatic evaluation of summaries, Text Summariz. Branches Out (2004).
[22] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, H. Li, Generative adversarial network for abstractive text summarization, 2018.
[23] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, H. Li, Generative adversarial network for abstractive text summarization, Association for the Advancement of Artificial Intelligence, 2018.
[24] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2873–2879.
[25] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y.-Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval, in: The 2015 Annual Conference of the North American Chapter of the ACL, 2015, pp. 912–921.
[26] Y. Lu, L. Liu, Z. Jiang, M. Yang, R. Goebel, A multi-task learning framework for abstractive text summarization (2019).
[27] Y. Luan, C. Brockett, B. Dolan, J. Gao, M. Galley, Multi-task learning for speaker-role adaptation in neural conversation models, in: Proceedings of the The 8th International Joint Conference on Natural Language Processing, 2017, pp. 605–614.
[28] M.-T. Luong, Q.V. Le, I. Sutskever, O. Vinyals, L. Kaiser, Multi-task sequence to sequence learning, in: International Conference on Learning Representations, 2016.
[29] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
[30] M. Nadejde, S. Reddy, R. Sennrich, T. Dwojak, M. Junczys-Dowmunt, P. Koehn, A. Birch, Predicting target language ccg supertags improves neural machine translation, in: Proceedings of the Second Conference on Machine Translation, Association for Computational Linguistics, Copenhagen, Denmark, 2017.
[31] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, 2016, pp. 280–290.
[32] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: The 40th Annual Meeting on Association for Computational Linguistics, 2002, pp. 311–318.
[33] R. Paulus, C. Xiong, R. Socher, A deep reinforced model for abstractive summarization, arXiv:1705.04304(2017).
[34] M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence level training with recurrent neural networks, in: The International Conference on Learning Representations (ICLR), 2016.
[35] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: bm25 and beyond, Found. Trends Inf. Retriev. 3 (4) (2009) 333–389.
[36] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 379–389.
[37] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2017, pp. 1073–1083.
[38] M. Steedman, J. Baldridge, Combinatory categorial grammar, Non-Transform. Syntax (2011) 181–224.
[39] R.S. Sutton, D.A. McAllester, S.P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: NIPS, 2000, pp. 1057–1063.
[40] J. Xu, X. Ren, J. Lin, X. Sun, Diversity-promoting gan: a cross-entropy based generative adversarial network for diversified text generation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3940–3949.
[41] M. Yang, Q. Qu, Y. Shen, K. Lei, J. Zhu, Cross-domain aspect/sentiment-aware abstractive review summarization by combining topic modeling and deep reinforcement learning, Neural Comput. Appl. (2018) 1–13.
[42] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: sequence generative adversarial nets with policy gradient., in: AAAI, 2017, pp. 2852–2858.
[43] L. Zhang, Y. Zhang, Y. Chen, Summarizing highly structured documents for effective search interaction, SIGIR, ACM, 2012.