Keyphrase Generation for Scientific Articles Using GANs (Student Abstract)

**Avinash Swaminathan,[1] Raj Kuwar Gupta,[1] Haimin Zhang,[2]
Debanjan Mahata,[2] Rakesh Gosangi,[2] Rajiv Ratn Shah[1]**
[1]MIDAS, IIIT-Delhi, India
[2]Bloomberg, USA
s.avinash.it.17@nsit.net.in, rajkuwargupta1996@gmail.com,
{hzhang449, dmahata, rgosangi}@bloomberg.net, rajivratn@iiitd.ac.in

Abstract

In this paper, we present a keyphrase generation approach using conditional Generative Adversarial Networks (GAN). In our GAN model, the generator outputs a sequence of keyphrases based on the title and abstract of a scientific article. The discriminator learns to distinguish between machine-generated and human-curated keyphrases. We evaluate this approach on standard benchmark datasets. Our model achieves state-of-the-art performance in generation of abstractive keyphrases and is also comparable to the best performing extractive techniques. We also demonstrate that our method generates more diverse keyphrases and make our implementation publicly available[1].

Introduction

Keyphrases are employed to capture the most salient topics of a long document and are indexed in databases for convenient retrieval. Researchers annotate their scientific publications with high quality keyphrases to ensure discoverability in large scientific repositories. Keyphrases could either be extractive (part of the document) or abstractive. Keyphrase generation is the process of predicting both extractive and abstractive keyphrases from a given document. This process is similar to abstractive summarization but instead of a summary the models generate keyphrases.

Researchers have achieved considerable success in the field of abstractive summarization using conditional-GANs (Wang and Lee 2018). There has also been growing interest in deep learning models for keyphrase generation (Meng et al. 2017; Chan et al. 2019). Inspired by these advances, we propose a new GAN architecture for keyphrase generation where the generator produces a sequence of keyphrases from a given document and the discriminator distinguishes between human-curated and machine-generated keyphrases.

Proposed Adversarial Model

As with most GAN architectures, our model also consists of a generator (G) and discriminator (D), which are trained in an alternating fashion (Goodfellow et al. 2014).

[1]Code available at https://github.com/avinsit123/keyphrase-gan

Generator - Given a document $d = \{x_1, x_2, ..., x_n\}$, where x_i is the i^{th} token, the generator produces a sequence of keyphrases: $y = \{y_1, y_2, ..., y_m\}$, where each keyphrase y_i is composed of tokens $y_i^1, y_i^2, ..., y_i^{l_i}$. We employ catSeq model (Yuan et al. 2018) for the generation process, which uses an encoder-decoder framework: the encoder being a bidirectional Gated Recurrent Unit (bi-GRU) and the decoder a forward GRU. To incorporate the out-of-vocabulary words, we use a copying mechanism (Gu et al. 2016). We also make use of attention mechanism to help the generator identify the relevant components of the source text.

Discriminator - We propose a new hierarchical-attention model as the discriminator, which is trained to distinguish between human-curated and machine-generated keyphrases. The first layer of this model consists of $m+1$ bi-GRUs. The first bi-GRU encodes the input document d as a sequence of vectors: $h = \{h_1, h_2, ..., h_n\}$. The other m bi-GRUs, which have the same weight parameters, encode each keyphrase as a vector: $\{k_1, k_2, ..., k_m\}$. We then use an attention-based approach (Luong, Pham, and Manning 2015) to build context vectors c_j for each keyphrase, where c_j is a weighted average over h. By concatenating c_j and k_j, we get a contextualized representation $e_j = [c_j; k_j]$ of keyphrase y_j.

The second layer of the discriminator is another bi-GRU which consumes the document representation h and the keyphrase representations e. The final state of this layer is passed through one fully connected layer (W_f) and sigmoid transformation to get the probability that a given keyphrase sequence is human-curated.

$$s_t = \begin{cases} GRU(h_t, s_{t-1}), \textbf{for } 1 \leq t \leq n \\ GRU(e_{t-n}, s_{t-1}), \textbf{ for } n+1 \leq t \leq n+m \end{cases}$$

$R(y_i) = D(y_i) = \sigma(W_f s_{i+n})$

GAN training - For a given dataset (S), which contain the documents and corresponding keyphrases, we first pretrain the generator (G) using Maximum Likelihood Estimation. We then use this generator to produce machine-generated keyphrases for all documents in S. These generated keyphrases along with the curated keyphrases are used to train the first version of the discriminator (D).

We then employ policy gradient reinforcement learning to train the subsequent versions of G. We freeze the weight

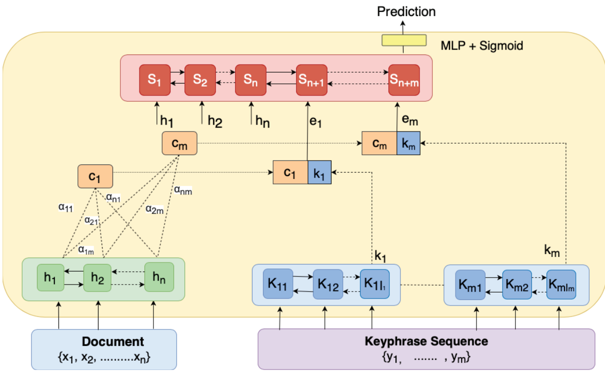

Figure 1: Schematic of Proposed Discriminator(D)

parameters of D and use it for reward calculation to train a new version of G. The reward for keyphrase is obtained from the last m states of the second bi-GRU layer in D (see Figure 1). The gradient update is given as:

$$\bigtriangledown R_G = \sum_{i=1}^{m} [D(y_i) - B] \bigtriangledown_{\phi} \phi(y^i)$$
$$\phi(y^i) = log \prod_{j=1}^{l_i} G(y_i^j | y_i^{1:j-1}, y_{1:i-1}, x)$$

where B is a baseline obtained by greedy decoding of keyphrase sequence. The resulting generator is then used to create new training samples for D. This process is continued till G converges.

Experiments and Results

We trained the proposed GAN model on KP20k dataset (Meng et al. 2017) which consists of 567,830 samples for training, 20,000 each for testing and validation. Each sample consists of an abstract, title, and the corresponding keyphrases of a scientific article. We evaluated the model on four datasets: Inspec, NUS, KP20k, and Krapivin, which contain 600, 211, 20,000, and 800 test samples respectively. For training G, we used Adagrad optimizer with learning rate ≈ 0.0005. We compare our proposed approach against 2 baseline models - catSeq (Yuan et al. 2018), RL-based catSeq Model (Chan et al. 2019) in terms of F1 scores as explained in (Yuan et al. 2018). The results, summarized in Table 1, are broken down in terms of performance on extractive and abstractive keyphrases.

For extractive keyphrases, our proposed model performs better than the pre-trained catSeq model on all datasets but is slightly worse than catSeq-RL except for on Krapivin where it obtains the best F1@M of 0.37. On the other hand, for abstractive keyphrases, our model performs better than the other two baselines on three of four datasets suggesting that GAN models are more effective in generation of keyphrases.

We also evaluated the models in terms of α-nDCG@5 (Clarke et al. 2008). The results are summarized in Table 2. Our model obtains the best performance on three out of the four datasets. The difference is most prevalent in KP20k, the largest of the four datasets, where our GAN model (at 0.85) is nearly 5% better than both the other baseline models.

Conclusion

In this paper, we propose new GAN architecture for keyphrase generation. The proposed model obtains state-of-the-art performance in generating abstractive keyphrases. To our knowledge, this is the first work that applies GANs to keyphrase generation problem.

Model	Score	Inspec	Krapivin	NUS	KP20k
Catseq(Ex)	F1@5	0.2350	0.2680	0.3330	0.2840
	F1@M	0.2864	0.3610	0.3982	0.3661
catSeq-RL(Ex.)	F1@5	**0.2501**	**0.2870**	**0.3750**	**0.3100**
	F1@M	**0.3000**	0.3630	**0.4330**	**0.3830**
GAN(Ex.)	F1@5	0.2481	0.2862	0.3681	0.3002
	F1@M	0.2970	**0.3700**	0.4300	0.3810
catSeq(Abs.)	F1@5	0.0045	0.0168	0.0126	0.0200
	F1@M	0.0085	0.0320	0.0170	0.0360
catSeq-RL(Abs.)	F1@5	0.0090	**0.0262**	0.0190	0.0240
	F1@M	0.0017	**0.0460**	0.0310	0.0440
GAN(Abs.)	F1@5	**0.0100**	0.0240	**0.0193**	**0.0250**
	F1@M	**0.0190**	0.0440	**0.0340**	**0.0450**

Table 1: Extractive and Abstractive Keyphrase Metrics

Model	Inspec	Krapivin	NUS	KP20k
Catseq	0.87803	0.781	0.82118	0.804
Catseq-RL	0.8602	**0.786**	0.83	0.809
GAN	**0.891**	0.771	**0.853**	**0.85**

Table 2: α-nDCG@5 metrics

References

Chan, H. P.; Chen, W.; Wang, L.; and King, I. 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *ACL*.

Clarke, C.; Kolla, M.; V. Cormack, G.; Vechtomova, O.; Ashkan, A.; Büttcher, S.; and MacKinnon, I. 2008. Novelty and diversity in information retrieval evaluation. *Proc. of the 31st ACM SIGIR* 659–666.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Meng, R.; Zhao, S.; Han, S.; He, D.; Brusilovsky, P.; and Chi, Y. 2017. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879*.

Wang, Y., and Lee, H. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. *CoRR* abs/1810.02851.

Yuan, X.; Wang, T.; Meng, R.; Thaker, K.; He, D.; and Trischler, A. 2018. Generating diverse numbers of diverse keyphrases. *ArXiv* abs/1810.05241.