Keyphrase Generation for Scientific Articles using GANs

Avinash Swaminathan†, Raj Kuwar Gupta†, Haimin Zhang$^{\delta}$, Debanjan Mahata$^{\delta}$, Rakesh Gosangi$^{\delta}$, Rajiv Ratn Shah †

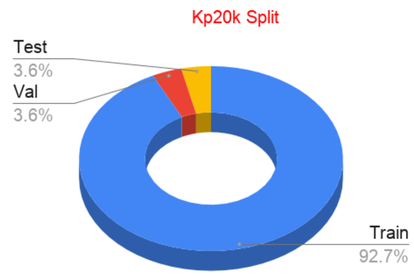
† MIDAS, IIIT-Delhi, India $^{\delta}$ Bloomberg, USA

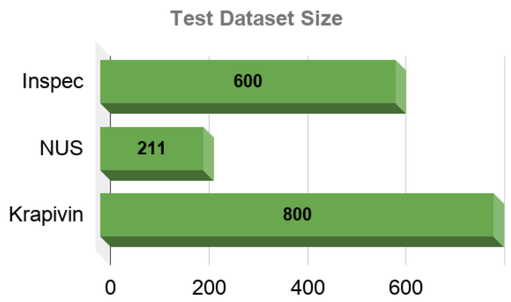
Research Contributions

- We propose a GAN model, consisting of a **generator** and **discriminator**, which generates keyphrase sequences from scientific articles
- The Generator **predicts a sequence of keyphrases** given the title and abstract of the scientific article
- The Discriminator **distinguishes between human-curated and machine-generated keyphrase sequences** and provides rewards for partially-decoded sequences
- We evaluate our approach on 4 standard datasets on the basis of F1 scores and diversity of keyphrases

Research Datasets

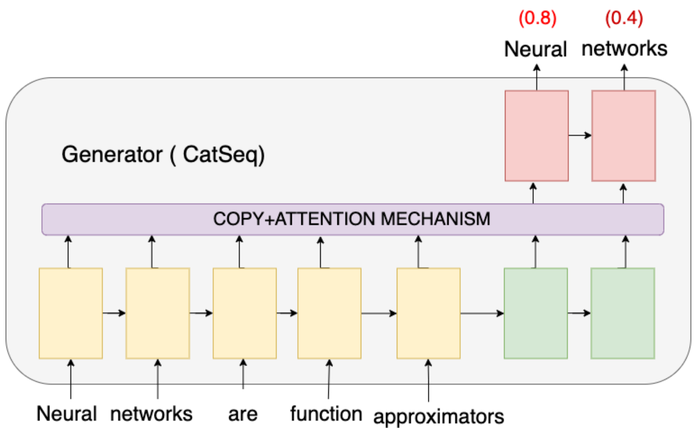
Train - Kp20k, **Val** - Kp20k, **Test** - Inspec, Krapivin, NUS, Kp20k





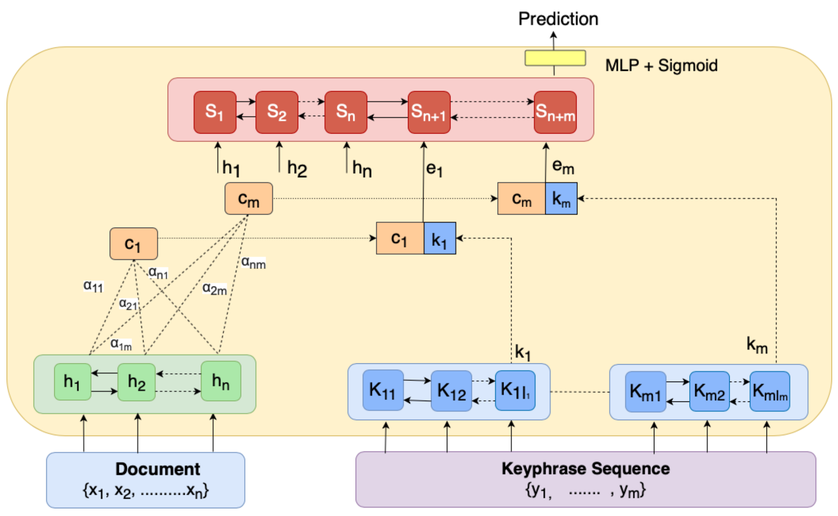
Generator

- A **Seq2Seq Encoder-Decoder model with attention mechanism** to identify relevant components of the source text and generate keyphrases
- It also incorporate **copy mechanism to copy words from source vocabulary** to overcome the limited target vocabulary.
- Given a document $d = \{x_1, x_2, ..., x_n\}$, where x_i is the ith token, the generator produces a sequence of keyphrases: $y = \{y_1, y_2, ..., y_m\}$, where each keyphrase y_i is composed of tokens $y_i^1, y_i^2, ..., y_i^{l_i}$



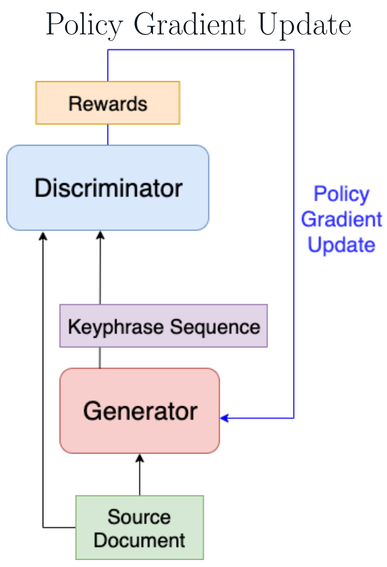
Discriminator

- Novel Discriminator based upon **hierarchichal-attention made up of 2 layers**.
- First Layer **generates a representation** for each keyphrase(k) and word(h) in the source document.
- Context vector(C) for each keyphrase built by calculating the **weighted average over source document representations**.
- C paired with k passed to 2nd layer alongside h to **compute probability keyphrase sequence is human-curated**

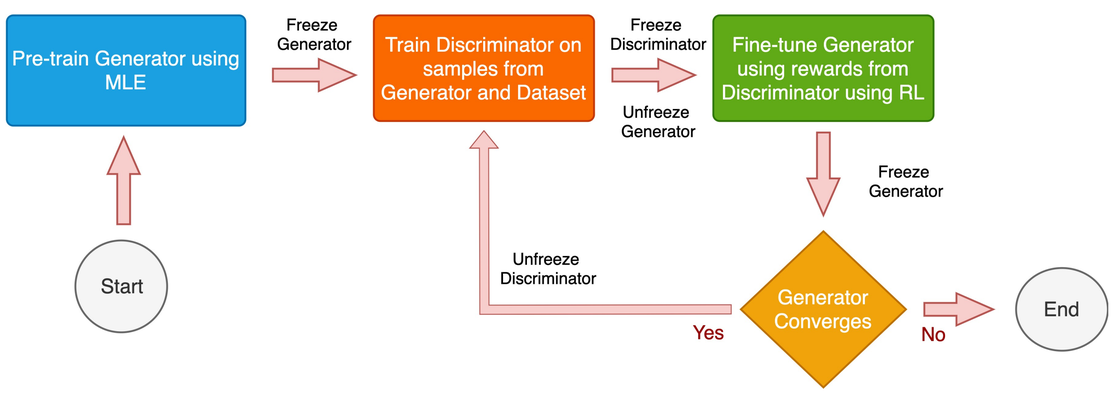


Policy Gradient

- **Discrete nature of text** makes it difficult to backpropagate through entire discriminator-generator setup
- Training the GAN using **policy gradient reinforcement learning** helps overcome this problem where scores from the discriminator are **rewards used to fine-tune the generator using RL**



Training Workflow



Equations

1. **Discriminator Loss**:Train Discriminator to distinguish between human-curated machine-generated keyphrase sequences

$$D_{loss} = -\mathbb{E}_{y \in S_r}[log(D(y))] - \mathbb{E}_{y \in S_f}[log(1 - D(y))]$$

2. **Discriminator Reward**: Reward for each keyphrase is the probability that it is human-curated. y_i denotes i^{th} keyphrase

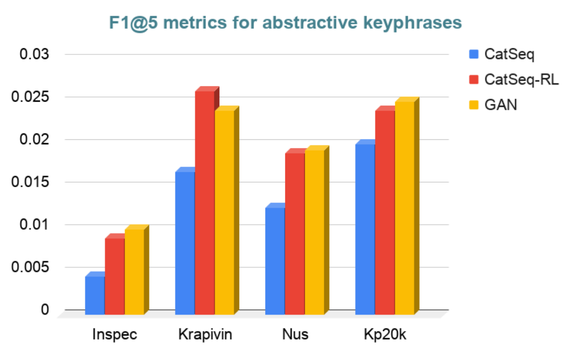
$$R(y_i) = D(y_i) = \sigma(W_f s_{i+n})$$

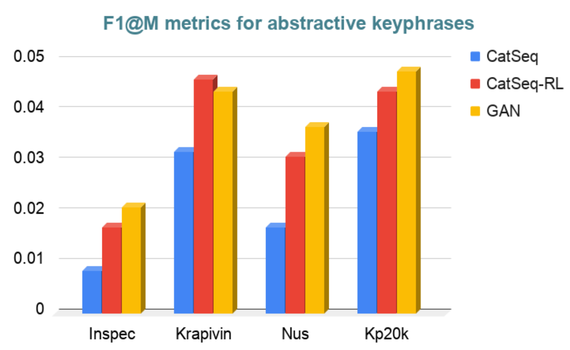
3. **Reinforcement learning**:Generator's policy is to maximize the expected reward of all the keyphrases it generates. The corresponding gradient update is

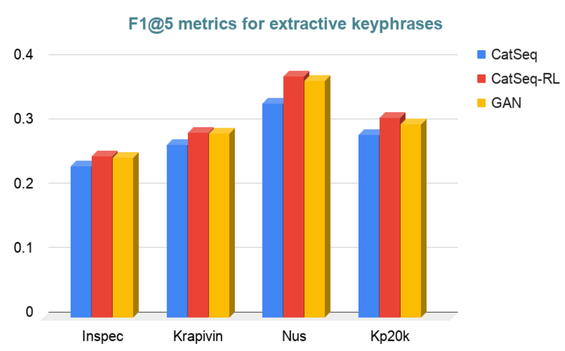
$$\nabla R_G = \sum_{i=1}^{m}[D(y_i) - B] \nabla log \prod_{j=1}^{l_i} G(y_i^j | y_i^{1:j-1}, y_{1:i-1}, x)$$

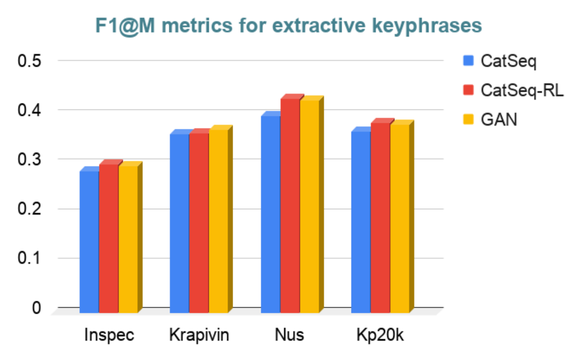
Experimental Results

- Results computed seperately for **abstractive** and **extractive** keyphrases
- Extractive keyphrases are **present in the source text**, while abstractive keyphrases are **inferred from context of the source text**
- Comparison Metrics include **F1@5 and F1@M** scores
- F1@5 computes F1 scores by **sampling top 5 keyphrases from generator**, while F1@M computes F1 score by taking **all keyphrases predicted by the generator** into account
- Proposed approach compared against **2 baseline models- CatSeq and CatSeq-RL**[1] for all test datasets
- GAN model performs **better than all baselines in case of abstractive keyphrases** and is **slightly worse than CatSeq-RL in case of extractive keyphrases**









References

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