# Keyword-constrained Natural Language Generation A Hybrid Approach using Supervised and Reinforcement Learning


**Sailik Sengupta***
Arizona State University
ssengu15@asu.edu

**He He, Batool Haider, Spandana Gella, Mona Diab**
Amazon AI
{hehea,bhaider,sgella,diabmona}@amazon.com



## ABSTRACT

Generating sentences that adhere to constraints has been an active area of research in natural language (eg. text summarizing, story generation) and computer vision (eg. image captioning, image to text). In this work, we consider the task of generating a sentence form an initial set-of-keywords– a ubiquitous *make-a-sentence* task in primary school classrooms. We adapt language models need to be adapted for this constrained setting– w reliance on sampling methods and reinforcement learning has been shown to result in high degrees of constraint satisfaction at the cost of generating low quality sentences. To mitigate these shortcomings, we propose a combined objective function that tries to mimic a language model while adhering to the imposed (keyword) constraints. We show how the use of a sequence-to-sequence model and policy gradient approach can be used to approximately minimize this objective and report scores that measure (1) the degree of constraint satisfaction and (2) the perplexity of the generated text.


## 1 INTRODUCTION AND RELATED WORK

It has been shown that a simple left-to-right language model (LM) given a prefix as input, is able to generate sentence completions. In the case of constrained text generation, LMs need to be account for inputs relating to the constraint(s). To this extent, one can use sequence-to-sequence architectures where the encoder encodes these additional inputs and the decoder is used this for left-to-right decoding. Such methods have been previously used for automatic story generation, where the topic is encoded and provided as input to a decoder [5]. For our problem, we are given a set of keywords $K = \{k_1, \ldots, k_n\}$ as input and are asked to generate a sentence or sequence of words $s = \langle w_1, \ldots, w_m \rangle$ such that $\forall\ k \in K, \exists\ i \in \{1, \ldots, m\}\ k = w_i$. Previous works have tried to solve this problem by using sampling methods that start with the set of keywords $K$ and in each sampling step, choose to insert, replace or delete a word with equal probability while ensuring that none of the keywords are deleted or replaced [4]. Sample are (accepted or) rejected based on a language model's score and a pre-defined threshold. Although this ensures that the generated sequences always contain the keywords, the generated sequences (1) have no well-defined stopping criterion, (2) can hardly be called an English sentence, (3) respect the sequence in which the keywords are arranged. Other methods like middle-out-decoding [3] can be leveraged if there $K$ is a singleton set but cannot be trivially applied otherwise.

*Work done while interning at Amazon AI.

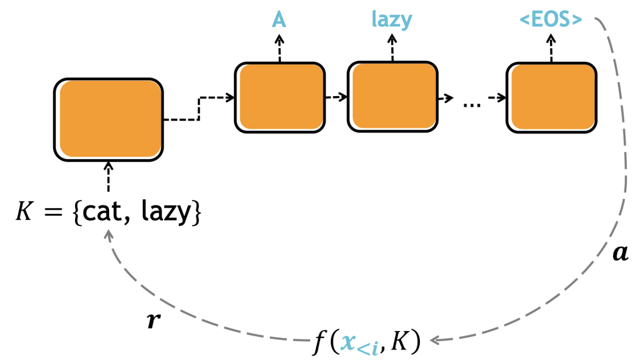


**Figure 1: Supervised learning creates a competent base model capable of handling sparse rewards at the end of a sentence. Reinforcement Learning enables us to fine-tune the model's weights and increase adherence to keyword constraints.**

## 2 APPROACH

Let us denote a hard scoring function $f(s, K) = 1$ iff all keywords in $K$ are present in $s$ and zero otherwise. Then a meaningful objective function for learning to generate sentence that includes all keywords would be,

$$\min_{\theta}\ \mathrm{KL}(p_\theta \| p_{\mathrm{LM}}) - \lambda \mathbb{E}_{p_\theta}[f(S, K)]$$

where $p_{\mathrm{LM}}$is a parametrized language model that can be used to generated sentences given a set of keywords and $p_\theta$ is a parameterized model, and $S$ is a sequence of words sampled from $p_\theta$ and $K$ is a sample of words sampled from the vocabulary. The hyper-parameter $\lambda$ allows us to trade-off between agreeing with the language model *vs.* satisfying constraints.

| Dataset | Model | Keyword Inclusion ↑ | Partial Inclusion ↑ | Perplexity ↓ | Vocabulary Diversity ↑ |
|---|---|---|---|---|---|
| MS-COCO Captions | | 1 | 1 | 177.057 | |
| | S2S | 0.52 | 0.10 | 81.88 | **0.28** |
| | S2S+attn | 0.72 | 0.84 | 89.19 | **0.28** |
| | S2S+attn+REINFORCE | **0.74** | **0.85** | **68.20** | 0.27 |
| Sentences for Translation | | 1 | 1 | 603.35 | |
| | S2S+attn | 0.70 | 0.83 | 568.59 | **0.37** |
| | S2S+attn+REINFORCE | **0.74** | **0.85** | **400.31** | 0.26 |

**Table 1: Keyword Inclusion metrics, perplexity and diversity in vocabulary of the generated sentences. ↑ (or ↓) indicate higher (or lower) values for that metric is better.**

In order to minimize this objective, we consider a two-fold approximation approach. First, we train a sequence-to-sequence (s2s) model (with attention) using supervised learning, i.e. a keywords to sentence model. This acts as a stand-in for minimizing the first term. Second, we fine tune the weights of this model using a policy gradient approach to optimize the second term. The choice of the number of episodes and the learning rate during the fine tuning phase act as a stand-in for the $\lambda$ term in the objective.

## 3 EXPERIMENTAL RESULTS

We consider the MS-COCO caption data-set [2] and the English sentences in the English to Turkish translation data-set [1]. After dividing it into training, validation and test set, we randomly sample two keywords from each sentence to generate the data for the supervised-learning phase. We use the validation set for fine-tuning using REINFORCE in the second phase. The results are shown in Table 1.

Using beam-search for decoding, we observe a high keyword inclusion rate after the supervised learning phase that further improves when fine tuned with policy gradient. Fortunately, we also notice a reduction in perplexity after fine-tuning, but we do notice a drop in the number of unique words generated in the output sentences drop after fine-tuning (esp for the sentences for the MT task).

Examples and additional experimental results can be found in our poster presented as the conference – https://tinyurl.com/md6t847w.

## REFERENCES


[1] Trang Ho. 2009. Tatoeba Project Blog.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[3] Shikib Mehri and Leonid Sigal. 2018. Middle-out decoding. In *Advances in Neural Information Processing Systems*. 5518–5529.

[4] Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2018. CGMH: Constrained Sentence Generation by Metropolis-Hastings Sampling. *arXiv preprint arXiv:1811.10996* (2018).

[5] Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. 2018. Plan-and-write: Towards better automatic storytelling. *arXiv preprint arXiv:1811.05701* (2018).