# Q-learning with Language Model for Edit-based Unsupervised Summarization

**Ryosuke Kohita** IBM Research
**Akifumi Wachi** IBM Research
**Yang Zhao** IBM Research
{kohi, akifumi.wachi, yangzhao}@ibm.com
**Ryuki Tachibana** IBM Research
ryuki@jp.ibm.com

## Abstract

Unsupervised methods are promising for abstractive textsummarization in that the parallel corpora is not required. However, their performance is still far from being satisfied, therefore research on promising solutions is on-going. In this paper, we propose a new approach based on Q-learning with an edit-based summarization. The method combines two key modules to form an Editorial Agent and Language Model converter (EALM). The agent predicts edit actions (e.t., delete, keep, and replace), and then the LM converter deterministically generates a summary on the basis of the action signals. Q-learning is leveraged to train the agent to produce proper edit actions. Experimental results show that EALM delivered competitive performance compared with the previous encoder-decoder-based methods, even with truly zero paired data (i.e., no validation set). Defining the task as Q-learning enables us not only to develop a competitive method but also to make the latest techniques in reinforcement learning available for unsupervised summarization. We also conduct qualitative analysis, providing insights into future study on unsupervised summarizers.[1]

## 1 Introduction

Automatic text summarization[2] is an attractive technique for helping humans to grasp the content of documents effortlessly. While supervised neural methods have shown good performances (See et al., 2017; Zhang et al., 2019), the unsupervised approach is starting to attract interest due to its advantage of not requiring costly parallel corpora. However, the empirical performance of unsupervised methods is currently behind that of state-of-the-art supervised models (Zhao et al., 2018; Baziotis et al., 2019). Unsupervised text summarization is still developing and is now at the stage where various solutions should be actively explored.

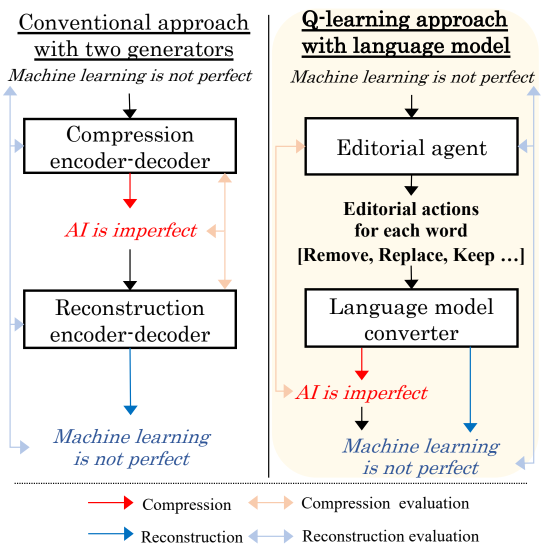


Figure 1: Overview of previous (left) and proposed (right) approaches on CR learning paradigm.

One previous unsupervised approach extends neural encoder-decoder modeling to the zero paired data scenario, where a model is trained with a paradigm called compression-reconstruction (CR) learning (Miao and Blunsom, 2016; Fevry and Phang, 2018; Zhao et al., 2018). The mechanism is similar to that of the back-translation (Sennrich et al., 2016): the model consists of a compressor (i.e., summarizer) and a reconstructor, and they are co-trained so that the reconstructor can recover the original sentence from the summary generated by the compressor (Miao and Blunsom, 2016; the left side of Figure 1). Experimental results showed that such an unsupervised encoder-decoder-based summarizer is able to learn the mapping from a sentence to a summary without paired data (Baziotis et al., 2019; Yang et al., 2020).

Reinforcement learning (RL) is also a potential

[1] Our codes are available at https://github.com/kohilin/ealm

[2] We refer to abstractive summarization in this paper.



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 470–484, November 16–20, 2020. 

solution for the no paired data situation. In related fields, for example, there are unsupervised methods for text simplification and text compression with policy-gradient learning (Zhang and Lapata, 2017; Zhao et al., 2018). Recent RL techniques take a value-based approach (e.g., Q-learning) such as DQN (Mnih et al., 2015) or the combination of policy and value-based approaches such as Asynchronous Advantage Actor-Critic (Mnih et al., 2016). A critical requirement to leverage a value-based method is a value function that represents the goodness of an action on a given state (Sutton et al., 1998). We can naturally define the value function by utilizing the CR-learning paradigm, and it makes the latest value-based approaches available for unsupervised text summarization.

In this paper, we propose a new method based on Q-learning and an edit-based summarization (Gu et al. 2019; Malmi et al. 2019; right side of Figure 1). The edit-based summarization generates a summary by operating an edit action (e.g., keep, remove, or replace) for each word in the input sentence. Our method implements the editing process with two modules: 1) an **E**ditorial **A**gent that predicts edit actions, and 2) a **L**anguage **M**model (LM) converter that deterministically decodes a sentence on the basis of action signals, which we call EALM. The CR learning is defined on the Q-learning framework to train the agent to predict edit actions that instruct the LM converter to produce a good summary. Although a vast action space causing sparsity in reward, such as the word generation of an encoder-decoder model, is generally difficult to be learned in RL, our method mitigates this issue thanks to its fewer edit actions and the deterministic decoding of a language model. Moreover, the formulation by Q-learning enables us to incorporate the latest techniques in RL.

The main contribution of this paper is that we provide a new solution in the form of an unsupervised edit-based summarization leveraging Q-learning and a language model. Experimental results show that our method achieved a competitive performance with encoder-decoder-based methods even with truly no paired data (i.e., no validation set), and qualitative analysis brings insights as to what current unsupervised models are missing. Also, the problem formulation on Q-learning enables us to import the latest techniques in RL, which leads to potential improvements in future research.

## 2 Task Definition

We begin by formally defining the problem of unsupervised summarization with the CR learning. The goal of the task is to produce an informative summary $\boldsymbol{y}$ consisting of $M$ words $y_1, y_2, ..., y_M$ for a given input sentence $\boldsymbol{x}$ consisting of $N$ words $x_1, x_2, ..., x_N$ where $M < N$. The challenge in this task is to learn the transformation from $\boldsymbol{x}$ to $\boldsymbol{y}$ with only the input sentence $\boldsymbol{x}$.

To tackle this, the CR learning introduces an additional transformation called reconstruction. The reconstruction requests to reproduce the input sentence $\hat{\boldsymbol{x}}$ from the generated summary $\boldsymbol{y}$ where $\hat{\boldsymbol{x}}$ is the reproduced sentence consisting of $N$ words $\hat{x}_1, \hat{x}_2, ..., \hat{x}_N$. In terms of the generated sentences $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$, let $C$ be the compression function and $R$ be the reconstruction function:

$$\boldsymbol{y} = C(\boldsymbol{x}, \theta_C) \,, \hat{\boldsymbol{x}} = R(\boldsymbol{y}, \theta_R) \,,$$

where $\theta_C$ and $\theta_R$ are their respective parameters. Thus, the task can be written as the following optimization problem:

$$\theta_C^*, \theta_R^* = \underset{\theta_C, \theta_R}{\arg\max} \{f(\boldsymbol{x}, \boldsymbol{y}) + g(\boldsymbol{x}, \hat{\boldsymbol{x}})\} \,,$$

where $f(\boldsymbol{x}, \boldsymbol{y})$ and $g(\boldsymbol{x}, \hat{\boldsymbol{x}})$ are functions to return a higher value for favorable $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$ in regard to the input sentence $\boldsymbol{x}$. According to the CR learning's hypothesis that the summary should contain enough information to guess the original contents, $\boldsymbol{y}$ becomes favorable when the difference between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is smaller while $\boldsymbol{y}$ maintains the essential aspects of a summary (e.g., shortness, fluency).

## 3 Previous Method

The previous approaches use a generative encoder-decoder model (Sutskever et al. 2014), for the compression and reconstruction functions (Miao and Blunsom, 2016; Fevry and Phang, 2018; Wang and Lee, 2018; Baziotis et al., 2019). Although the objective functions and implementation details differ depending on the study, the underlying motivation entails the same hypothesis as the CR learning. For example, Baziotis et al. (2019) introduced four objective functions — discrepancy of $\boldsymbol{y}$ from a pre-trained language model, topical distance of $\boldsymbol{x}$ and $\boldsymbol{y}$, and the length of $\boldsymbol{y}$ and probability difference of $x_i$ and $\hat{x}_i$ — where the former threes can be regarded as $f(\boldsymbol{x}, \boldsymbol{y})$ and the final one as $g(\boldsymbol{x}, \hat{\boldsymbol{x}})$.

While such an encoder-decoder model has performed well on many generation tasks, it suffers from inherent difficulties related to repetition (See et al., 2017), length control (Kikuchi et al., 2016), and exposure bias (Ranzato et al., 2016). It also runs into convergence problems when co-training multiple generators (Salimans et al., 2016).

# 4 Proposed Method

Our proposed method, which we call EALM, consists of two essential modules: the editorial agent and the LM converter. The agent sends action signals (keep, remove, or replace each word in a sentence) to the conveter, which then deterministically transforms the input sentence according to the signals. We train the agent to find action signals so that the LM converter produces sentences demanded by the CR learning. In the following sections, we first share the background of Q-learning (§4.1) and then present how to put the task and our approach on the Q-learning framework (§4.2). We next explain the core algorithmic details (§4.3) and finish with explanations about training and inference (§4.4).

## 4.1 Preliminaries

Q-learning is a popular approach in RL as represented by Deep Q-Networks (DQN, Mnih et al. 2015). Q-learning leverages an action-value function to estimate the *value* of a pair of state and action with respect to a policy $\pi$. The action-value function (i.e., Q-function) is represented as the expected reward for the state-action pair:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right],$$

where $s$ is a state, $a$ is an action, $r$ is a reward function for the state-action pair, and $\gamma$ is the discount factor. Hence, to solve a text summarization task via Q-learning, we first need to appropriately define the state, action, and reward function.

## 4.2 Unsupervised Edit-based Summarization with Q-learning

In our approach, given an input sentence $\boldsymbol{x}$, we define a state $s_i$ in regard to each word $x_i$. An action $a_i$ for the state $s_i$ is chosen from among the three options, $\mathcal{A} = \{\texttt{Remove}, \texttt{Keep}, \texttt{Replace}\}$. The goal of the editorial agent is to provide the optimal action sequence, $\boldsymbol{a} = \{a_0, a_1, \ldots, a_N\}$, by iteratively making action decisions on each word (§4.3.1). To obtain $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$, we propose a deterministic transformation algorithm based on $\boldsymbol{a}$ and the LM converter (§4.3.2). Finally, we define the reward function $r$ to evaluate the action and action sequence in terms of the produced sentences $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$ (§4.3.3). The reward function is designed to align with the CR learning paradigm and leads the agent into bringing the action sequence that generates an appropriate $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$.

## 4.3 Algorithms

In this section, we describe three principle algorithms: 1) how to create $s_i$ and to predict $a_i$, 2) how to generate $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$ by means of $\boldsymbol{a}$ and the LM converter, and 3) how to compute the reward.

### 4.3.1 Iterative Action Prediction

The overall flow of iteratively predicting an action for each word is shown in Figure 2. The agent predicts an action for a state (i.e., a word) one by one, so we call one prediction a *step* and express it with a subscription $(t)$. For example, $s_{i(t)}$ and $a_{i(t)}$ respectively denote the state and action for $x_i$ at the $t$-th step. Note that $a_{i(t)}$ has a predicted action if the agent has already done the prediction on $x_i$ by the $t$-th step, otherwise $a_{i(t)}$ is $\texttt{Keep}$. Also, we prepare a Boolean vector $\boldsymbol{u}_{(t)}$ of length $N$ representing the prediction statuses at the $t$-th step; $u_{i(t)}$ is 1 if the prediction on the $i$-th word has been finished, otherwise, 0. The order to predict an action is determined by Q-values. Let $s^*$ and $a^*$ be a state-action pair to be operated next, which comes from the maximum Q-values over unoperated states:

$$s^*, a^* = \mathop{\arg\max}_{s \in \mathcal{S}', a \in \mathcal{A}} Q(s, a),$$

where $\mathcal{S}'$ is defined as $\mathcal{S}' = (\forall_i)\{s_{i(t)} \mid u_{i(t)} = 0\}$. The agent then reiterates the predictions until it finishes determining an action on all words. By defining the state in regard to a single word instead of a whole sentence and asking the agent to determine the prediction order, we can handle variable sentence lengths in natural language. Note that this is not a left-to-right process; the agent conducts the prediction in the order of "confidence".

Next we explain how to encode $s_{i(t)}$. To send the agent contextual information, such as the previous decisions, the prediction statuses, and the whole sentence, we dynamically create a state $s_{i(t)}$ with a concatenation of two encodings; *local encoding* $\boldsymbol{l}_{i(t)}$ and *global encoding* $\boldsymbol{g}_{i(t)}$

$$s_{i(t)} = [\boldsymbol{l}_{i(t)}; \boldsymbol{g}_{i(t)}].$$

To create the two encodings, first, we map $x_i$ to a fixed-sized vector $\boldsymbol{e}_i$ with an arbitrary encoder (we

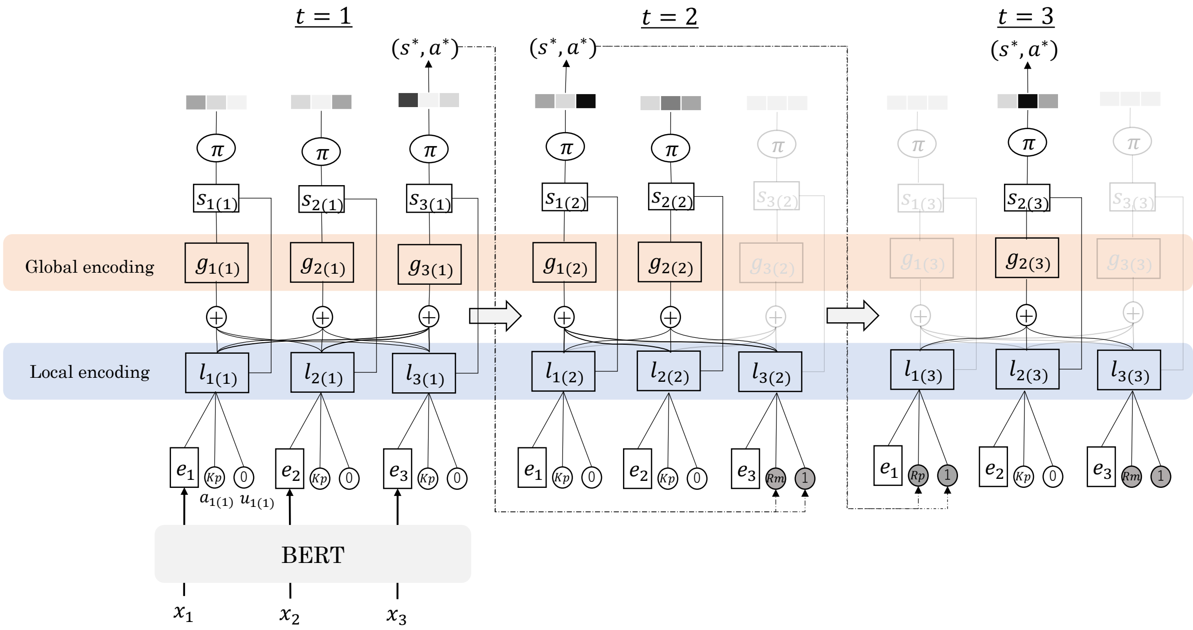


Figure 2: Algorithmic visualization of iterative action prediction

use BERT; Devlin et al. 2019), and $\boldsymbol{e}_i$ is repeatedly used throughout the process regardless of the steps. Then, we define the local encoding as

$$\boldsymbol{l}_{i(t)} = \boldsymbol{e}_i + \boldsymbol{b}^{a_{i(t)}} + \boldsymbol{b}^{u_{i(t)}},$$

where $\boldsymbol{b}^{a_{i(t)}}$ and $\boldsymbol{b}^{u_{i(t)}}$ are learnable bias vectors for the action and prediction status of the $i$-th word, respectively. Next, we create the global encoding in a self-attention fashion as

$$\boldsymbol{g}_{i(t)} = \sum_j w_{j(t)} \boldsymbol{l}_{j(t)},$$

where $w_{j(t)}$ is computed with ReLU:[3]

$$w_{j(t)} = \frac{\mathrm{ReLU}(\boldsymbol{l}_{i(t)} \cdot \boldsymbol{l}_{j(t)})}{\sum_k \mathrm{ReLU}(\boldsymbol{l}_{i(t)} \cdot \boldsymbol{l}_{k(t)})}.$$

Thanks to the bias terms in $\boldsymbol{l}_{i(t)}$ and the self-attention in $\boldsymbol{g}_{i(t)}$, $s_{i(t)}$ is aware of the previous decisions for each word and the interactions between those decisions. In addition, BERT encoding $\boldsymbol{e}_i$ enables us to take a whole sentence into account.

### 4.3.2 Deterministic Decoding by Language Model with Action Signals

In this section, we explain how to compress and reconstruct sentences in a deterministic manner

[3]We used ReLU$(\cdot)$ instead of the conventional $\exp(\cdot)$ because $\exp(\cdot)$ caused the exploding gradient in our case.

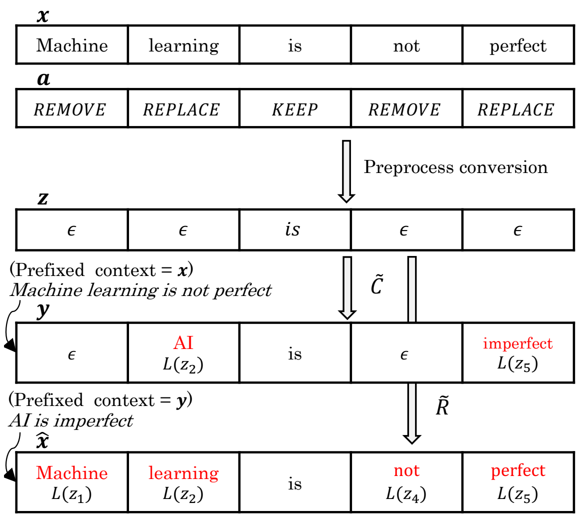


Figure 3: Deterministic compression and reconstruction with masked language model

with the LM converter. For the LM converter, we use BERT (Devlin et al., 2019) which is a masked language model (MLM) trained to predict "masked" portions in a sentence. MLM can estimate the probability distribution of $i$-th word $x_i$ in a sentence as $p(x \mid \boldsymbol{x}_{\setminus i})$ where $\boldsymbol{x}_{\setminus i}$ is the same as $\boldsymbol{x}$ except that it has a mask at the $i$-th position ($\langle ..., x_{i-1}, [\texttt{MASK}], x_{i+1}, ...\rangle$; Wang and Cho 2019). $L(\boldsymbol{x}_{\setminus i})$ denotes a function to return a word with the highest probability for the $i$-th position.

The procedure to obtain $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$ by using $\boldsymbol{a}$

and MLM is shown in Figure 3. First, we convert $\boldsymbol{x}$ to a skeleton sequence $\boldsymbol{z}$ consisting of $N$ tokens $z_1, z_2, ..., z_N$ where $z_i$ is $x_i$ if $a_i$ is `Keep`, otherwise a null token $\epsilon$. We then define our compression and reconstruction functions $\tilde{C}$ and $\tilde{R}$ as

$$
\begin{aligned}
y_i &= \tilde{C}(\boldsymbol{z}, a_i, L) \\
&= \begin{cases} L(\boldsymbol{z}_{\setminus i}) & (a_i = \texttt{Replace}) \\ z_i & (\text{otherwise}), \end{cases} \\
\hat{x}_i &= \tilde{R}(\boldsymbol{z}, a_i, L) \\
&= \begin{cases} L(\boldsymbol{z}_{\setminus i}) & (a_i \in \{\texttt{Remove}, \texttt{Replace}\}) \\ z_i & (\text{otherwise}). \end{cases}
\end{aligned}
$$

A word is predicted only for $\epsilon$ given by `Replace` in compression, but it does so for all $\epsilon$ in reconstruction. Also, we set the original sentence as a prefixed context, which comes from $\boldsymbol{x}$ in compression and $\boldsymbol{y}$ in reconstruction, to make MLM aware of a former meaning. An example is shown in Figure 3, where MLM receives "*Machine learning is not perfect* . [MASK] *is* [MASK] ." as the compression input and predicts words for the [MASK]s. If there are multiple masks, we conduct the prediction in an autoregressive fashion (see Appendix A.1). Note that while any language model can be used for the LM converter, MLM is advantageous because it utilizes before and after contexts, and there is no restriction on looking ahead at upcoming words.

#### 4.3.3 Stepwise Reward Computation

In this section, we explain the reward computation of the chosen action by referring to $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$.

As stated in §4.3.1, we have an action sequence $\boldsymbol{a}_{(t)}$ for every step $t$. When we apply $\tilde{C}$ and $\tilde{R}$ to all the $\boldsymbol{a}_{(t)}$, we can obtain a list of tuples $(s, a, r, \boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{x}})_{(t)}$. A tuple — let us say, *experience* — enables us to evaluate a state-action pair with respect to a single transition. In this section, we propose three techniques — **step reward**, **violation penalty**, and **summary assessment** — to evaluate the agent's behavior with the stepwise experiences. Refer to Table 1 to see how these work in reward computation with an actual example.

Before moving on to the details, let us define two important notions throughout this section, compression rate ($cr$) and reconstruction rate ($rr$):

$$
cr_{(t)} = 1 - \frac{|\boldsymbol{y}_{(t)}|}{|\boldsymbol{x}|}, \; rr_{(t)} = \frac{|\{i \mid x_i = \hat{x}_{i(t)}\}|}{|\boldsymbol{x}|}.
$$

The CR learning assumes that the higher values of $cr$ and $rr$ are better. We use these for calculating rewards and pruning experiences.

**Step Reward.** The task of the agent is to produce an action sequence with which the LM converter generates an appropriately compressed sentence while keeping the reconstruction successful. As such, we define the reward function $r$ as

$$
r(s, a, \boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{x}}) = r_{SR} + r_{SA},
$$

where $r_{SR}$ is the step reward that are designed to encourage the agent to improve the compression and reconstruction rate, respectively. $r_{SA}$ is an additional score from the qualitative assessment of $\boldsymbol{y}$, which we explain later. Returning to the step reward $r_{SR}$, it is a multiplication of $r_C$ and $r_R$ defined as

$$
r_{SR} = r_C \times r_R,
$$

$$
r_C = 1 - \frac{|\boldsymbol{y}_{(t)}|}{|\boldsymbol{y}_{(t-1)}|}, \; r_R = \begin{cases} 1 & (rr_{(t)} > \tau_{(t)}) \\ -1 & (\text{otherwise}) \end{cases},
$$

where $\tau_{(t)}$ is a minimum requirement for the reconstruction rate at the $t$-th step and is defined as $\tau_{(t)} = 1 - t\frac{1-\tau}{N}$ with the hyperparameter $\tau \in [0, 1]$. If we set $\tau = 1$ that requests perfect reconstruction, then $\tau_{(t)} = 1$ regardless of $t$. However, we need to forgive reconstruction failure to some extent because of the information loss in compression, and $\tau$ adjusts the allowed number of failures. For example, $\tau = 0.5$ requests the model to recover at least half of the original sentence correctly.

Let us describe the behavior of the step reward $r_{SR}$. First, the reward is 0 when the agent chooses `Keep` or `Replace` because $r_C = 0$ due to there being no change in the length of $\boldsymbol{y}$. Second, the reward gets a positive value when the agent chooses `Remove` and satisfies the requirement for the reconstruction rate ($rr_{(t)} > \tau_{(t)}$). Third, the reward gets a negative value when the agent chooses `Remove`, but the reconstruction rate is less than the requirement. In short, the step reward recommends `Remove` as long as the agent can recover the original word, and otherwise, `Keep` or `Replace`.

**Violation Penalty.** Sequential modeling, including that performed by our agent, essentially suffers from error propagation caused by incorrect predictions at an earlier stage (Collins and Roark, 2004). The violation penalty mitigates this issue by giving a negative reward to the latest problematic action and excluding experiences after the mistake.

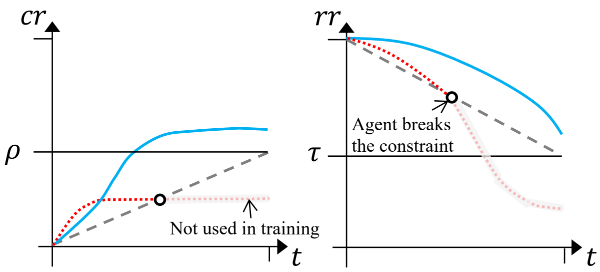


Figure 4: Violation penalty for compression (left) and reconstruction (right). The x-axis is step and the y-axis is each ratio. The horizontal lines in the middle are $\rho$ and $\tau$, and the dashed lines represent $\rho_{(t)}$ and $\tau_{(t)}$. The circles represent a step where the agent breaks the constraints.

Here, in addition to $\tau$, we introduce the hyperparameter $\rho$, which represents a minimum requirement for the compression rate. $\rho_{(t)}$ denotes its threshold at the $t$-step defined as $\rho_{(t)} = t\frac{\rho}{N}$, and the agent must satisfy the condition $cr_{(t)} > \rho_{(t)}$. As the penalty, we forcibly assign $-1$ reward for the state-action pair at the $T$-th step when the agent breaks either constraint of $\tau_{(T)}$ or $\rho_{(T)}$. In addition, we ignore experiences from step $(T+1)$ and onward. If the agent keeps predicting until the end, we define $T = N$. Figure 4 shows how these constraints work for the experience sequence.

**Summary Assessment.** Although the step reward considers the compression and reconstruction ratios, it ignores the critical aspects of the generated summary such as replacement with a shorter synonym and fluency as a sentence. Here, we explain the $r_{SA}$ mentioned in the previous paragraph and describe how to reflect such qualitative assessments to the reward given to the agent.

As the essential properties for $\boldsymbol{y}$, we take three perspectives into account: informativeness, shortness, and fluency. The informativeness refers to how much $\boldsymbol{y}$ retains the original meaning of $\boldsymbol{x}$, and the shortness and fluency are self-explanatory. To reflect these perspectives onto the agent's decision, we define $r_{SA}$ as

$$r_{SA} = \frac{T}{N} \cdot [cr_{(T)} \times rr_{(T)} + \alpha \cdot sim(\boldsymbol{x}, \boldsymbol{y}_{(T)}) + \beta \cdot llh(\boldsymbol{y}_{(T)})],$$

where $sim$ computes a similarity score of $\boldsymbol{x}$ and $\boldsymbol{y}$, and $llh$ computes a log-likelihood of $\boldsymbol{y}$. $\alpha$ and $\beta$ are hyperparameters to adjust the importance of $sim$ and $llh$. In addition to $r_{SR}$, we give $r_{SA}$ to the experiences from the beginning to $T$-th steps as defined in the step reward paragraph.

Let us explain the terms inside the square brackets first. The first term, which is the multiplication of $cr_{(T)}$ and $rr_{(T)}$, aims for shortness and informativeness. It gets a higher value when the agent achieves the right balance of compression and reconstruction. The second term $sim$ aims to evaluate informativeness brought about by `Replace`. Concretely, $sim$ returns a semantic similarity score in the range of $[0, 1]$ through the sentence vectors of $\boldsymbol{x}$ and $\boldsymbol{y}_{(T)}$ rather than just checking exact matches of words. The last term $llh$ represents fluency via the log-likelihood of $\boldsymbol{y}_{(T)}$ given by a pre-trained language model (Zhao et al., 2018). We use BERT for the computation of $sim$ and $llh$ (Devlin et al. 2019; Wang and Cho 2019; see Appendix A.3). Finally, $T/N$ is the ratio of the number of operated words. It becomes closer to 1 when the agent is reaching a termination, i.e., finishing the prediction on all words by avoiding the violation penalty, which makes $r_{SA}$ larger. In contrast, the agent who fails at an earlier stage gets a small value of $r_{SA}$.

### 4.4 Training and Inference

**Training.** Leveraging the experiences $(s, a, r, \boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{x}})$ in the replay buffer (Lin, 1992), the agent learns the policy for summarizing a sentence $\boldsymbol{x}$ within the Q-learning framework. Specifically, we utilize DQN (Mnih et al., 2015) to learn the Q-function $Q^*$ corresponding to the optimal policy by minimizing the loss,

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}[(Q^*(s, a) - \psi)^2],$$

where $\psi = r + \gamma \max_{a'} \bar{Q}^*(s', a')$ and $\bar{Q}$ is a target Q-function whose parameters are periodically updated in accordance with the latest network parameters. During the collection of experiences, RL requires the agent to explore an action on a given state for finding a better policy. As a unique point in this work, the agent must explore not only the action but also the order to predict. For both explorations, we use the $\epsilon$-greedy algorithm (Watkins, 1989) that stochastically forces the agent to ignore Q-values and to behave randomly (see Appendix A.2).

**Inference.** Our modeling that provides $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$ for each step has another advantage in terms of the inference. For the final output, we use $\boldsymbol{y}$ at the $t^*$-th step that achieves the best balance of the compression and reconstruction ratios, where $t^* =$

| t | Action | Type | 1 | 2 | 3 | 4 | 5 | 6 | $cr/\rho_{(t)}$ | $rr/\tau_{(t)}$ | $\frac{T}{N}$/$cr$$rr$/$sim$/$llh$ | $r_{SR}$ / $r_{SA}$ | $r$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | `Remove` (May) | $\boldsymbol{y}_{(1)}$ | | the | force | be | with | you | .17/.05 | 1.0/.91 | - | 1.0 / .20 | 1.2 |
| | | $\hat{\boldsymbol{x}}_{(1)}$ | May | the | force | be | with | you | | | | | |
| 2 | `Remove` (the) | $\boldsymbol{y}_{(2)}$ | | | force | be | with | you | .33/.10 | 1.0/.83 | - | 1.0 / .20 | 1.2 |
| | | $\hat{\boldsymbol{x}}_{(2)}$ | May | the | force | be | with | you | | | | | |
| 3 | `Remove` (force) | $\boldsymbol{y}_{(3)}$ | | | | be | with | you | .50/.15 | .50/.75 | .50/.25/.50 /1.0 | **-1.0** / .20 | -.80 |
| | | $\hat{\boldsymbol{x}}_{(3)}$ | I | will | always | be | with | you | | | | | |
| 4<br>5<br>6 | No experiences due to the violation occurred at the step 3. | | | | | | | | | | | | |

Table 1: An example of stepwise reward computation. It breaks the reconstruction constraint at the step 3 when removing *force*, so $r_{SR} = -1$. $r_{SA}$ is computed at the step 3 by $0.5 \times (0.25 + 0.5 \times 0.1 + 1.0 \times 0.1) = 0.20$, and it is used for the step 1 and 2 as well. The settings of hyperparameters are $\tau = 0.5$, $\rho = 0.3$, $\alpha = 0.1$, and $\beta = 0.1$.

$\arg\max_t\{cr_{(t)} + rr_{(t)}\}$. This is based on the trade-off relationship of compression and reconstruction as seen in the precision-recall curve.

## 5 Experiment

**Baselines.** We compare our proposed approach with three baselines: Lead-N, which simply takes the beginning N words as the summary, SEQ3, a recent encoder-decoder model (Baziotis et al., 2019), and CMatch, a new approach without explicit reconstruction learning (Zhou and Rush, 2019). To conduct qualitative analysis on generated summaries, we ran the baselines ourselves with a replicated model for SEQ3[4] and the provided model for CMatch.[5] Also, we test two types of SEQ3 models: one tuned with a validation set (SEQ3$^+$) and the other with parameters at the last iteration in the training (SEQ3$^-$). This is because EALM and CMatch do not need paired data even for validation.

**Proposed method.** We implemented EALM as follows. The Q-network of the agent consists of a two-layered MLP with 200 units per layer and ReLU. We used the Adam optimizer with the learning rate of 0.001 and apply gradient clipping by 1. For the epsilon-greedy strategy, we first set the exploring probability to 0.9 and decay it by multiplying by 0.995 every 100 updates until it reaches the minimum exploration rate of 0.03. We set the discount factor $\gamma$ to 0.995. The size of the replay buffer is 2000, and we sample 128 experiences as a batch for one update. As the final model, we use parameters at a time when the averaged score of reward in the replay buffer is maximum, i.e., our model does not need a validation set. The hyperparameters of step reward ($\tau$, $\rho$; §4.3.3) are set to 0.5 and 0.3, respectively. The hyperparameters of summary assessment ($\alpha$, $\beta$; §4.3.3) are both set to 0.1. We train three models with the same configuration and report their averaged score, as Q-learning inherently contains randomness in training.

**Dataset.** The same as Baziotis et al. (2019), we train our model on the Gigaword corpus (GIGA, Rush et al., 2015). However, we used only $30K$ sentences randomly picked from sentences with less than 50 words for the training of EALM. This is because the whole data, $3.8M$ sentences, is too large to expose the agent to different experiences from the same sentence.[6] Note that we used the entirety of sentences for the training of the SEQ3 models.

We followed Baziotis et al. (2019) in the evaluation as well, using the test set consisting of the GIGA (1897 sentences) and DUC datasets (DUC3 with 624 sentences, DUC4 with 500 sentences; Over et al. 2007).

All models follow the same tokenization policy: the default tokenization in GIGA, DUC3, and DUC4. Although BERT (which EALM uses) has its own vocabulary based on subwords, we do not apply subwording to go along with a single tokenization policy. Therefore, words not in the BERT vocabulary are interpreted as unknown words, and the ratio of unknown words was around 10% in GIGA.

[4] `https://github.com/cbaziotis/seq3`. We ran the training script with the same configuration as the original paper except for decreasing the batch size from 128 to 32 due to our GPU limitation. We trained three models and obtained slightly lower scores than the ones reported in the original paper. We report the averaged score among the three models.

[5] `https://github.com/jzhou316/Unsupervised-Sentence-Summarization`

[6] EALM can be trained with the large dataset, but it takes long time due to the exploitation and exploration learning strategy of Q-learning. 30K was better in the balance of the required time and the model performance.

| Data & Model | | R-1 | R-2 | R-L | LEN | NW |
|---|---|---|---|---|---|---|
| GIGA | L8 | 21.78 | 7.62 | 20.40 | 8.00 | 0 |
| | L15 | 24.22 | 8.20 | 22.00 | 15.00 | 0 |
| | S3$^+$ | 23.15 | 7.56 | 21.11 | 14.77 | 0.59 |
| | S3$^-$ | 22.09 | 6.59 | 20.02 | 14.63 | 1.09 |
| | CM | **26.71** | **10.12** | **24.67** | 9.48 | 0.44 |
| | EL | 25.00 | 7.61 | 22.48 | 17.39 | 0.07 |
| DUC3 | L8 | 18.34 | 5.76 | 16.92 | 8.00 | 0 |
| | L15 | 20.94 | **6.20** | 18.54 | 15.00 | 0 |
| | S3$^+$ | 20.09 | 5.53 | 17.76 | 16.51 | 0.71 |
| | S3$^-$ | 19.57 | 5.17 | 17.25 | 16.42 | 1.17 |
| | CM | 17.50 | 4.84 | 16.35 | 5.18 | 0.39 |
| | EL | **21.69** | 5.25 | **18.88** | 19.61 | 0.02 |
| DUC4 | L8 | 18.85 | 4.88 | 17.05 | 8.00 | 0 |
| | L15 | 22.14 | **6.25** | 19.30 | 15.00 | 0 |
| | S3$^+$ | 21.69 | 5.87 | 18.81 | 16.81 | 0.59 |
| | S3$^-$ | 21.25 | 5.64 | 18.32 | 16.69 | 1.08 |
| | CM | 18.62 | 5.60 | 17.16 | 5.26 | 0.36 |
| | EL | **22.50** | 5.80 | **19.47** | 20.46 | 0.01 |

Table 2: ROUGE scores, averaged lengths (LEN), and averaged occurrences of new words (NW). L8 and L15 are Lead-N. S3$^{[+-]}$ represent SEQ3 models. CM is CMatch and EL is EALM. ROUGE scores are computed with summaries capped at the first 75 bytes.

**Evaluation.** In our quantitative analysis, we examine the ROUGE scores.[7] To mitigate the bias to longer sentences in ROUGE calculation, we capped all summaries at the first 75 bytes. Note that the averaged sentence length of gold summaries after the capping were 8.58, 9.59, and 10.25 for GIGA , DUC3, and DUC4, respectively. Also, we examine sentence length (LEN) and count of new words (NW; number of words that are used in a generated summary but do not appear in the input sentence). Additionally, we show qualitative comparisons with a manual check of generated summaries. Although a questionnaire survey is often conducted to assess the deeper quality of summaries such as informativeness and readability, this still hides the exact points of model's strengths and weaknesses. We consider that specific indications provide insights on future work for the current unsupervised summarizers. We manually checked more than 200 summaries for each model and each dataset and include a few samples in Appendix (A.6).

**Results.** Table 2 lists the results of ROUGE scores, averaged lengths, and averaged counts of new words. EALM showed a better performance in DUC3 and DUC4 with respect to R-1 and R-L. In GIGA, it performed competitively with the baselines. However, the original length of the generated summaries tended to be longer, and the occurrence of new words was the lowest.

[7]We used files2rouge (`https://github.com/pltrdy/files2rouge`) following Baziotis et al., 2019.

| | | |
|---|---|---|
| S3$^+$ | ✔ | Grammatical |
| | ✔ | Informative |
| | ✘ | Copy words from the top as it is |
| | ✘ | Meaningless rephrasing |
| CM | ✔ | Grammatical |
| | ✔ | Fluent in successful cases (in GIGA) |
| | ✘ | Lack of information (in DUC3 and DUC4) |
| | ✘ | Too much short (in DUC3 and DUC4) |
| EL | ✔ | Select words from the whole input |
| | ✔ | Contain keywords |
| | ✘ | Less grammatical |
| | ✘ | Lack of rephrasing |

Table 3: Pros (✔) and cons (✘) found in the generated summaries of SEQ3, CMatch, and EALM.

CMatch achieved the highest scores of ROUGE and meaningful length in GIGA. The scores of R-2 and R-L were superior to others by about two points, which means CMatch captured not only salient words but also word co-occurrences. However, for generating summaries, CMatch uses a language model trained with gold summaries in GIGA. In other words, it may just internally store the probable word distributions in summary sentences on GIGA. Actually, the results on DUC3 and DUC4 were not better than those on GIGA. Even though CMatch does not require paired data, it is not practical to collect enough summaries to train a language model for each domain.

SEQ3 showed a competitive performance with other models, but its scores dropped when no validation set was available. The requirement of a validation set is a keen disadvantage because creating input-summary pairs comes at a significant human labor cost.

While almost all of the best scores were given by the statistical models, Lead-15 also performed competitively. This result indicates that unsupervised summarization methods can not yet overcome the trivial baseline. One significant barrier preventing the progress of unsupervised methods is presumably the difficulty of rephrasing. For writing a good summary, condensing a longer expression into a shorter form is essential. As seen in the NW column in Table 2, the number of new words was less than one in SEQ3$^+$, CMatch, and EALM. The current models tend to operate just by copy-and-paste, which is consistent with the report by Baziotis et al. (2019).

Finally, we manually assess the summaries produced by each model and sum up their pros and cons in Table 3. Also, actual examples are shown in

| INPUT | Human | SEQ3 | CMatch | EALM |
|---|---|---|---|---|
| japan 's nec corp. and **UNK** computer corp. of the united states said wednesday they had agreed to join forces in supercomputer sales . | nec UNK in computer sales tie-up | japan 's nec corp. and *her* computer corp. of the united states said | nec *agrees* to join forces in supercomputer sales | nec computer united states said agreed *(to)* join forces in sales |
| mechanical problems **that** threaten to shut down the **astronomical** observations of the hubble space telescope may **prompt a repair mission** six months earlier than planned to the $ 1.7 billion spacecraft , nasa officials told congress on wednesday . | Problems may stop Hubble astronomical observations; NASA may **accelerate repair mission** | mechanical problems that threaten to shut down the *her* observations of the hubble space telescope threaten | *nasa observes* | *that* threaten to shut down astronomical observations of space telescope may *prompt repair mission* six earlier than planned billion spacecraft *nasa officials told congress on* |
| endeavour 's astronauts connected the first two **building blocks of the international space station** on sunday , **creating a seven-story tower in the shuttle cargo bay** . | First 2 **building blocks of international space station** successfully joined. | endeavour 's astronauts connected the first two *building blocks of the international space station* | endeavour 's astronauts create a shuttle | connected first *building blocks of international space* on *creating tower in shuttle bay* |

Table 4: Summaries by Human (gold reference), SEQ3, CMatch and EALM from GIGA (top), DUC3 (center), and DUC4 (buttom).

Table 4. First, we found that a summary of SEQ3 was likely to be an exact copy of the input sentence from the top, but it kept sentences grammatical and informative. Rephrasing by SEQ3 did not meet our expectation in most cases, such as changing a week of the day (e.g., *Wednesday* to *Thursday*) or a common adjective to a pronoun adjective (e.g., *astronomical* to *her*). CMatch stably generated fluent summaries in GIGA, as seen in the ROUGE scores. It also generated grammatically correct sentences such as number agreement (e.g., *nec agrees* ...). In the DUC datasets, however, meaningless summaries increased, such as containing no important information (e.g., *nasa observes*). Relatedly, CMatch's summaries on DUC3 and DUC4 were too short, and we found that more than half of the summaries consisted of less than or qeual to 5 words. Finally, EALM's outputs tended to be longer due to containing non-informative portions (e.g., *nasa officials told* ...). It was also likely to be ungrammatical due to leaving only a functional word (e.g., *mechanical problems that threaten* ...) or deleting required prepositions (e.g., ... *agreed (to) join* ...). Those failures resulted in lower readability. However, EALM tried to keep keywords from the whole input even though they exist at latter positions in a sentence, which is also supported by the relatively higher score of R-1 and R-L. Although this *challenge* caused low readable and ungrammatical summaries, it is an interesting research direction to sophisticate such EALM's behavior.

## 6 Conclusion

We brought the Q-learning framework into unsupervised text summarization and proposed a new method EALM that is an edit-based unsupervised summarizer leveraging a Q-learning agent and a language model. The experments showed that EALM performed competitively with the previous encoder-decoder-based methods. However, in qualitative analysis, we found that the quality of the generated summaries of any unsupervised model was not sufficient, and there are individual limitations for each model. These issue must be overcome as the step forward to generating practically available summaries without paired data. In particular for EALM, there is room for improvement by importing the latest techniques in RL. Our work paves the way for further research on bridging Q-learning and unsupervised text summarization.

## References


Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQ^3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 673–681.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 111–118.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising autoencoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Proceedings of 32th International Conference on Neural Information Processing Systems*, pages 11179–11189.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.

Long Ji Lin. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 319–328.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pages 1928–1937.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of 4th International Conference on Learning Representations*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved techniques for training gans. In *Proceedings of the 29th Neural Information Processing Systems*, pages 2234–2242.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3104–3112.

Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.

Yaushian Wang and Hung-Yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187–4195.

Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards*. Ph.D. thesis, King's College, Cambridge, UK.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. Ted: A pretrained unsupervised summarization model with theme modeling and denoising.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China. Association for Computational Linguistics.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa. 2018. A language model based evaluator for sentence compression. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 170–175.

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106, Florence, Italy. Association for Computational Linguistics.

## A Appendices

### A.1 Autoregressive Prediction with MLM

Algorithm 1 describes the autoregressive prediction with MLM, which we used when an input contains multiple masks.

**Algorithm 1** Autoregressive prediction with MLM

**Input:** a sentence $\boldsymbol{x}$ that includes [MASK]s
**Outpit:** a sentence $\boldsymbol{x}$ after replacing all [MASK]s with predicted words
$I \leftarrow (\forall i)\{i \mid x_i = \text{[MASK]}\}$
**while** $I \neq \phi$ **do**
  **for** $j \in I$ **do**
    $w_j \leftarrow L(\boldsymbol{x}_{\setminus j})$
  **end for**
  $j^* \leftarrow \arg\max_{j \in I} P(w_j | x_{\setminus j})$
  $x_{j^*} \leftarrow w_{j^*}$
  $I \leftarrow I_{\setminus j^*}$
**end while**

### A.2 Exploration of Prediction Order

As explained in section 4.3 in the main paper, the editorial agent explores the order to predict. While the agent basically chooses a state with a maximum Q-value as the next state, we sometimes pick a most uncertain state instead. We define the uncertainty of a state by the entropy of action probabilities as $H(s) = -\sum_{a \in \mathcal{A}} Q(s,a) \log Q(s,a)$, and then $s^*$ and $a^*$ are selected as

$$s^* = \arg\max_{s \in s^t} H(s)\ , \ a^* = \arg\max_{a \in \mathcal{A}} Q(s^*, a)\ .$$

### A.3 Semantic Similarity and Log-likelihood Computation in Summary Assessment

**Semantic Similarity.** We use a pre-trained model to predict the semantic similarity of paired-sentences with their BERT encodings.[8] The model is trained in a supervised manner with a pair of sentences and their similarity score. The original library outputs a real-valued score in the range of $[0, 5]$, whereas we normalize it to $[0, 1]$.

**Log-likelihood.** We compute the log-likelihood of a compressed sentence by using BERT as follows (Wang and Cho, 2019):

$$\frac{1}{M} \sum_{i \in M} \log(P(y_i \mid \boldsymbol{y}_{\setminus i}))\ .$$

[8] https://github.com/AndriyMulyar/semantic-text-similarity

However, our $llh$ function performs thresholding — namely, it returns 1 if the score is beyond a threshold, otherwise 0 — because the raw log-likelihood score is not scaled with the other rewards. We empirically set the threshold to 0.005.

### A.4 Relaxations in $rr_{(t)}$ Calculation

The calculation of the reconstruction rate introduced in section 4.3.3 is based on an exact match of each word of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. Given the ambiguity of natural language, this is very strict, so the agent rarely acquires rewards. We relax this situation by 1) excluding stop words in the calculation and 2) comparing with top-k candidates. Therefore, the equation of $rr$ can be formally re-written as

$$rr_{(t)} = \frac{|\{i \mid x_i \in L^k(\boldsymbol{z}_{\setminus i}) \cap x_i \notin W\}|}{|\{i \mid x_i \notin W\}|}\ ,$$

where $L^k(\boldsymbol{z}_{\setminus i})$ returns top-k probable words for the $i$-th position and $W$ is a pre-defined set of words. We set $k = 10$. We used common stopwords (e.g., *him*, *the*) and infrequent words in GIGA for $W$.

### A.5 Experimental Details

**Computing Infrastructure.** We run the models on a machine with the below specifications:

- Ubuntu 18.04
- Intel(R) Xeon(R) @ 2.60GHz
- RAM 120GB
- NVIDIA Tesla P100

**Model Size.** In EALM, the number of trainable parameters was 348208 in our experimental setting, which is all for the editorial agent. There are no trainable parameters for the language model.

**Hypperparameter Search.** We did not conduct a hyperparameter search. We empirically determined the values described in the main paper (the “Proposed method” paragraph in §5).

**Runtime Speed.** EALM processes a sentence in three seconds pm average on the above GPU.

### A.6 Generated Summaries

Samples of the summaries generated by each model are listed in the tables on the following next pages. These examples are taken from the first sentences for GIGA and randomly picked for DUC3 and DUC4. We also include human-generated summaries (i.e., gold reference).

| INPUT | Human | SEQ3 | CMatch | EALM |
|---|---|---|---|---|
| japan 's nec corp. and UNK computer corp. of the united states said wednesday they had agreed to join forces in supercomputer sales . | nec UNK in computer sales tie-up | japan 's nec corp. and her computer corp. of the united states said | nec agrees to join forces in supercomputer sales | nec computer united states said agreed join forces in sales |
| the sri lankan government on wednesday announced the closure of government schools with immediate effect as a military campaign against tamil separatists escalated in the north of the country . | sri lanka closes schools as war escalates | the sri lankan government on thursday announced the closure of government schools with immediate military country | sri lankan government announces military campaign against tamil separatists | sri lankan government announced closure government schools effect as military campaign escalated north country |
| police arrested five anti-nuclear protesters thursday after they sought to disrupt loading of a french antarctic research and supply vessel , a spokesman for the protesters said . | protesters target french research ship | police arrested five anti-nuclear protesters tuesday after they sought to disrupt her of antarctic protesters | police arrest five anti-nuclear protesters | police arrested after sought disrupt loading of french antarctic research supply vessel spokesman for said |
| factory orders for manufactured goods rose #.# percent in september , the commerce department said here thursday . | us september factory orders up #.# percent | factory orders for manufactured goods rose #.# percent in september | factory orders rise #.# percent in september | factory orders manufactured goods rose september commerce said here |
| the bank of japan appealed to financial markets to remain calm friday following the us decision to order daiwa bank ltd. to close its us operations . | bank of UNK UNK for calm in financial markets | the bank of japan appealed to financial markets to remain calm thursday following decision | the bank of daiwa ltd. to close its us operations | bank japan appealed financial markets to remain calm following us decision order bank to close us operations |
| croatian president franjo tudjman said friday croatian and serb negotiators would meet saturday to thrash out an agreement on the last serb-held area in croatia , under a deal reached at us-brokered talks . | rebel serb talks to resume saturday : tudjman by peter UNK | croatian president franjo tudjman said thursday croatian and serb negotiators would meet saturday to agreement talks | croatian president franjo tudjman says serb negotiators will meet | croatian said croatian serb would meet thrash out an agreement on last area croatia under deal reached at talks |
| japan 's toyota team europe were banned from the world rally championship for one year here on friday in a crushing ruling by the world council of the international automobile federation -lrb- fia -rrb- . | toyota are banned for a year | japan 's toyota team europe were banned from the world rally championship for one here fia | europe is banned from the world championship for one year | japan toyota team europe banned from world rally championship for year here in crushing ruling council international automobile . |

Table 5: Summaries by Human (gold reference), SEQ3, CMatch and EALM from GIGA,

| INPUT | Human | SEQ3 | CMatch | EALM |
|---|---|---|---|---|
| mechanical problems that threaten to shut down the astronomical observations of the hubble space telescope may prompt a repair mission six months earlier than planned to the $ 1.7 billion spacecraft , nasa officials told congress on wednesday . | Problems may stop Hubble astronomical observations; NASA may accelerate repair mission | mechanical problems that threaten to shut down the her observations of the hubble space telescope threaten . | nasa observes | that threaten to shut down astronomical observations of space telescope may prompt repair mission six earlier than planned billion spacecraft nasa officials told congress on |
| perhaps no city offers a more public example of the problems of homelessness than san francisco , the biggest complaint visitors lodge about the city concerns the aggressive panhandling and other manifestations of homelessness that they experience , say city tourist officials . | Lack of affordable housing basic to San Francisco's homeless crisis. | perhaps no city offers a more public example of the problems of her than san offers more public tourist | san francisco city lidge | perhaps no city offers more public example of problems of than san francisco , complaint lodge about city concerns aggressive and other of that experience , say city tourist officials |
| atlanta – maybe , just maybe , customers who pay to use bank atm machines are beginning to fight back , or maybe they 're just getting smarter . | Bank customers beginning to resist double charges on ATM use. | atlanta – maybe , just maybe , customers who pay to use bank machines getting | atlanta gets smarter | atlanta maybe , just maybe customers who pay to use bank atm machines beginning fight , or maybe getting smarter |
| the head of turkey 's pro-islamic party said thursday he would not insist on his rightful chance to lead turkey 's next government , heading off a confrontation with the military that would only deepen the nation 's political crisis . | Broad-based secularist coalition likely in Turkey. | the head of turkey 's her party said tuesday he would not insist on rightful turkey 's crisis . | the head of turkey 's pro-islamic party | head of turkey party said he would insist rightful chance to lead turkey next government heading off confrontation with military that would only nation political crisis |
| suicide bombers targeted a crowded open-air market friday , setting off blasts that killed the two assailants , injured 21 shoppers and passersby and prompted the israeli cabinet to put off action on the new peace accord . | Possible early detonation of car bomb still injures 21, bombers killed | suicide bombers targeted a crowded open-air market tuesday , setting off blasts that killed assailants accord . | israeli cabinet puts off accord on | suicide bombers targeted crowded market setting off blasts that killed two , injured 21 and and prompted the israeli cabinet to put off action on new peace accord |
| president nelson mandela acknowledged saturday the african national congress violated human rights during apartheid , setting him at odds with his deputy president over a report that has divided much of south africa . | President Nelson Mandela acknowledges ANC rights violations. Other leaders disagree. | president clinton mandela acknowledged saturday the african national congress violated human rights during apartheid setting africa | nelson mandela acknowledges human rights | mandela acknowledged national congress violated human during setting at odds with deputy president over report that divided much of south |

Table 6: Summaries by Human (gold reference), SEQ3, CMatch and EALM from DUC3.

| INPUT | Human | SEQ3 | CMatch | EALM |
|---|---|---|---|---|
| endeavour 's astronauts connected the first two building blocks of the international space station on sunday , creating a seven-story tower in the shuttle cargo bay . | First 2 building blocks of international space station successfully joined. | endeavour 's astronauts connected the first two building blocks of the international space station | endeavour 's astronauts create a shuttle | connected first building blocks of international space on creating tower in shuttle bay |
| in a cocoon of loyal and wealthy supporters , president clinton said friday that he must " live with the consequences " of his mistakes , although he contended that democrats should take pride in the achievements of his presidency and take heart from its possibilities . | Clinton supports candidates, speaks at fundraisers, acknowledges mistakes. | in a her of loyal and wealthy supporters , president clinton said tuesday that must of loyal and wealthy supporters , clinton " . | democrats take pride in presidency | in a of loyal and wealthy supporters president clinton said that must live with consequences of mistakes although he that should pride in achievements of presidency and heart from possibilities |
| on the eve of a holiday that has been linked to antiabortion violence , the authorities on tuesday were investigating whether a picture of an aborted fetus sent to a canadian newspaper was connected to last month 's fatal shooting of a buffalo , n.y. doctor who provided abortions or four similar attacks in western new york and canada since 1994 . | Anti-abortion flyer in Canada may be related to Buffalo clinic slaying | on the eve of a holiday that has been linked to her violence , authorities of holiday that has been linked to her violence , | on the eve of a holiday | on eve of holiday that has been linked to violence on investigating picture of an sent to canadian newspaper was connected to last month fatal shooting of buffalo , doctor who provided or similar attacks in western new york and canada since 1994 |
| famine-threatened north korea 's harvest will be no better this year than last and could be worse , a senior u.n. aid official said saturday . | World Food Program reports famine may have killed 2 million North Koreans | her north korea 's harvest will be no better this year than last worse | south korea 's zhan | north korea harvest better last could worse senior aid official said |
| matthew wayne shepard , the gay student who was beaten in the dead of night , tied to a fence and left to die alone , was mourned at his funeral friday by 1,000 people , including many who had never met him . | Matthew Shepard eulogized as one who wanted to make people's lives better | matthew wayne her , the gay student who was beaten in the dead of night , gay student who was beaten him | us ceos | , gay who beaten in dead of night tied to fence and left to die alone was at funeral by including who had met |
| a delegation of chilean legislators lobbying against the possible extradition of augusto pinochet to spain to face trial , warned thursday that chile was on the brink of political turmoil . | Chilean legislators protest in Madrid against extradition of Pinochet | a delegation of chilean legislators lobbying against the possible extradition of augusto pinochet to face turmoil | delegation of chilean legislators faces trial | delegation of chilean legislators lobbying against possible of augusto spain to trial , warned chile on brink of political turmoil |

Table 7: Summaries by Human (gold reference), SEQ3, CMatch and EALM from DUC4.