Learning to Extract Coherent Summary via Deep Reinforcement Learning

Anonymous authors

Abstract

Coherence plays a critical role in producing a high-quality summary from a document. In recent years, neural extractive summarization is becoming increasingly attractive. However, most of them ignore the coherence of summaries when extracting sentences. As an effort towards extracting coherent summaries, we propose a neural coherence model to capture the cross-sentence semantic and syntactic coherence patterns. The proposed neural coherence model obviated the need for feature engineering and can be trained from scratch in an end-to-end fashion using a large corpus. Empirical results show that the proposed neural coherence model can effectively capture the cross-sentence coherence patterns. Using the output of the neural coherence model and ROUGE as the reward, we design a reinforcement learning method to train a proposed neural extractive summarizer which is named Reinforced Neural Extractive Summarization (RNES) model. The RNES learns to optimize coherence and informative importance of the summary simultaneously. Experiment results show that the proposed RNES outperforms existing baselines and achieves the state-of-art performance in term of ROUGE on the CNN/Daily Mail dataset. The qualitative evaluation indicates that the summary produced by the proposed RNES is more coherent and readable.

Introduction

Although deep neural networks (DNN) have dominated almost every field of natural language processing, such as sentiment classification (Tang, Qin, and Liu 2015), machine translation (Bahdanau, Cho, and Bengio 2014) and question answering (Zhou et al. 2015), generating high-quality summaries from long documents is still a very challenging task. Most of the recent works on abstractive summarization focus on headline generation from one paragraph (Rush, Chopra, and Weston 2015) or several sentences (Hu, Chen, and Zhu 2015) by using sequence-to-sequence architectures borrowed from neural machine translation. However, they bypass the fundamental problems in summarization, namely the representation of long documents and the generation of multiple coherent sentences. These models fail to produce readable, informative and coherent sentences when dealing with long documents. There is still a long way to go before abstractive summarization becomes practicable.

In contrast, extracting sentences from documents to form summaries, also named extractive summarization, is a more practical approach, because it can guarantee the grammatical correctness of the produced summary and its semantic relevance with the corresponding document. Extractive summarization has been researched for several decades. Traditional methods mainly focus on scoring sentences using graph-based method (Erkan and Radev 2004), submodular functions (Lin and Bilmes 2011) or integer linear programming (Berg-Kirkpatrick, Gillick, and Klein 2011), which are coupled with handcrafted features. As the distributed representation shows its outstanding capability in capturing semantic and syntactic information of text (Mikolov et al. 2013), there is an emergence of works that use the deep neural networks to extract salient sentences (Cheng and Lapata 2016; Nallapati, Zhai, and Zhou 2017). Although DNN-based methods can identify the important sentences from the documents, they still lack the ability to ensure coherence of the summary. They may produce summaries with sentences that are semantically independent to each other, which would cause difficulty for readers to comprehend the story as a whole.

The coherence of a summary is essential for its readability and clarity. However, to the best of our knowledge, there is no work incorporating coherence into the neural extractive model while extracting sentences. This task is challenging because it is difficult to include coherence into the objective function of supervised learning models because the coherence also depends on sentences that are eventually extracted when the inference is performed. In contrary, reinforcement learning (RL) is suitable for this case. RL algorithms aim to train an agent to maximize the reward by interacting with an environment. It is usually used in cases that the objective is not differentiable with respect to the model parameters. RL has been used in sequence generation tasks such as abstractive summarization generation (Paulus, Xiong, and Socher 2017) and neural machine translation (Nguyen, Boyd-Graber, and Daume 2017).

In this paper, we focus on improving the coherence ability of neural extractive model via reinforcement learning. We need a model that estimates the coherence in the first place. During the past decades, works in coherence modeling mainly focus on topical coherence. The most popular method is the entity grid model (Barzilay and Lapata 2008)

which constructs a grid to represent grammatical and semantic transitions of entities between sentences. The entity grid is then converted to discrete feature vectors which are used as the input to a learning model for predicting the coherence score (Nguyen and Joty 2017). Models based on entity grid require the entity labels, whose accuracy depends on the performance of the named entity recognition algorithm, which may become the bottleneck of these methods. Furthermore, entity grid models transitions of different entities separately, so it fails to capture semantic correlation between entities. Therefore, we instead use a neural coherence model which learns to estimate the coherence degree between two sentences by their distributed representation in an end-to-end fashion.

The contribution of this paper is twofold. First, we propose a novel neural coherence model which uses a low-dimension dense distributed representation of texts instead of sparse discrete features. The proposed neural coherence model does not rely on any entity recognition systems and can be trained from scratch in an end-to-end fashion. The neural coherence model can capture the cross-sentence local entity transition and discourse relation with multiple layers of convolution and max-pooling. The neural coherence model is trained with a large-scale unlabeled corpus. The experiment results show that, given one sentence, the neural coherence model can effectively identify the appropriate next sentence to compose a coherent sentence pair.

Second, we design a novel Reinforced Neural Extractive Summarization (RNES) model that incorporates coherence into neural extractive summarization with reinforcement learning. The output of the neural coherence model is used as immediate rewards during the training of RNES so that it learns to extract more coherent summaries. ROUGE score is utilized as the final reward, and hence the proposed RNES model finds a balance between coherence and informative importance of sentences. We evaluate the proposed RNES model on CNN/Daily Mail dataset, and the results show that it achieves the state-of-the-art performance on ROUGE metrics. The qualitative evaluation indicates that the summaries produced by RNES are more informative and coherent.

Related Work

Our research builds on previous works in the field of neural extractive summarization, reinforcement learning, and coherence modeling.

Much progress has been made beyond traditional frameworks of extractive summarization models. Most of the recent works are based on deep neural networks and distributed representation of text, which regards the extractive summarization as a sequence labeling problem. For example, (Filippova et al. 2015) use a recurrent neural network (RNN) to delete words from a sentence for sentence compression task. (Cheng and Lapata 2016) use a convolutional neural network to encode sentences, and then an RNN reads the sentence representations sequentially to encode the document. Finally, another RNN is used to label sentences sequentially, taking the encoded document representation and the previously labeled sentences into account. (Cheng and Lapata 2016) mainly consider the importance of sentences and the non-redundancy of the summary. (Nallapati, Zhai, and Zhou 2017) use a similar architecture to encode document, but it explicitly models sentence content, salience, novelty and position in its model for extracting sentences.

Our work is also related to the application of reinforcement learning on document summarization. Different from document classification whose target is a given ground truth label, the goal of machine translation or automatic text summarization may be multiple potential sequences of words. Such tasks are natural candidate problems for reinforcement learning by using some evaluation metrics to judge the quality of the result, such as BLEU, ROUGE, etc. Deep reinforcement learning has drawn considerable attention in recent years. For example, (Paulus, Xiong, and Socher 2017) use the ROUGE-L score as a reinforcement reward and self-critical policy gradient training algorithm to train an abstractive summarization models. (Ayana et al. 2016) and (Ranzato et al. 2015) show that directly optimizing the evaluation metrics ROUGE via reinforcement learning is more effective than optimizing likelihood for the sequence generation. All the works above focus on abstractive summarization by optimizing ROUGE metrics, and to date, no work has been done to apply RL to neural extractive summarization with coherence as the reward. To our knowledge, our work is the first step toward filling this gap.

An essential requirement for summarization systems is the coherence of its output. Coherence is what makes multiple sentences semantically, logically and syntactically coherent (Yao, Wan, and Xiao 2017). Not surprisingly, a variety of coherence models have been developed over the years, among which the entity grid model proposed by (Barzilay and Lapata 2008) is the most popular one. However, the discrete representation of entity grid suffers from the curse of dimensionality problem which limits its application on text summarization. (Nguyen and Joty 2017) presented a local coherence model based on a convolutional neural network that operates over the distributed representation of entity grid. But entity grid is limited in modeling coherence relationship between different entities. Since (Nguyen and Joty 2017) still rely on the entity grid features, it fails to exploit the full power of DNN in learning the hidden distributed representation of text. (Li and Hovy 2014) use the recurrent or recursive neural network to obtain the distributed representation of sentences and then use a pairwise ranking method to train the coherence model. (Li and Hovy 2014) select the window of sentences from original articles as positive examples and randomly replace some of the sentences in the window to form negatives examples. This model does not need any feature engineering, but it is weak in capturing the local entity transition because of the lack of cross-sentence local interaction. Our neural coherence model can be trained from scratch in an end-to-end fashion. It can model the local entity transitions as well as the syntactic and semantic relation between sentences via different levels of cross-sentence local interaction.

Neural Extractive Summarization Model

We need to construct a neural extractive summarization model (NES) before training it with reinforcement learning algorithms. In this section, we present the detailed architecture of our NES.

The extractive summarization model reads the document and selects a set of sentences sequentially as a summary. Given a document $X = (x_1, x_2, \cdots, x_n)$ that consist of n sentences, the neural extractive summarization models aim to output a sequence of binary decisions $Y = (y_1, y_2, \cdots, y_n)$, where n denotes the number of sentences in the document and $y_i \in \{0, 1\}$ indicates whether sentence x_i is selected. Then the extracted summary is a sequence of l sentences denoted as

$$G = \text{extract}(X, Y) = (x_{q_1}, \cdots, x_{q_l}),$$

where $1 \le q_1 < \cdots < q_l \le n$, and $y_{q_i} = 1$ for $i = 1, \cdots, l$.

The NES uses a hierarchical neural network to encode the document. At the word-level, convolutional neural network (CNN) is used to extract features of words and their context. Let $x_t = (w_1, w_2, \cdots, w_m)$ denotes the t-th sentence with m words, and v denotes the size of word embedding. Then the sentence could be represented by a matrix $\mathbf{M} \in \mathbb{R}^{m \times v}$. Multiple convolution kernels with different kernel size are used to extract features of word w_i:

$$\mathbf{f}_i^j = \mathbf{M}_{i:i+k_j-1} \odot \mathbf{W}_j + \mathbf{b}_j,$$

where $\odot$ represents element-wise product, $\mathbf{W}_j, \mathbf{b}_j, k_j$ are the kernel weight matrix, the bias and the kernel size of the j-th convolution kernel respectively. The word w_i is represented by concatenating the feature maps $\mathbf{f}_{w_i} = [\mathbf{f}_i^1; \mathbf{f}_i^2; \cdots]$. The representation of sentence x_t is represented as the mean of all its word features

$$\mathbf{x}_t = \frac{1}{m} \sum_{i=1}^{m} \mathbf{f}_{w_i}.$$

At the sentence-level, we use a bi-directional gated recurrent unit (Bi-GRU) to model the context of sentences. Gated recurrent unit is a variant of recurrent neural network proposed by (Chung et al. 2014). It has two gates, an update gate $\mathbf{z}_t$ and reset gate $\mathbf{r}_t$. The hidden state $\mathbf{h}_t$ at time step t could be computed with following equations:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{V}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{V}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \\
\hat{\mathbf{h}}_t &= \tanh(\mathbf{W}_h \mathbf{x}_t + \mathbf{V}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h), \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \hat{\mathbf{h}}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1},
\end{aligned}$$

where $\mathbf{W}_*$'s, $\mathbf{V}_*$'s and $\mathbf{b}_*$'s are parameters of GRU.

Using Bi-GRU, the representation of the t-th sentence $\mathbf{x}_t$ is transformed to a forward hidden state $\overrightarrow{\mathbf{h}}_t$ and a backward hidden state $\overleftarrow{\mathbf{h}}_t$. Both states are are concatenated to form the contextual representation of the t-th sentence $\overleftrightarrow{\mathbf{h}}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$. The entire document is represented as $\mathbf{d}$ by a nonlinear transformation of the mean-pooling output:

$$\mathbf{d} = \tanh(\mathbf{W}_d (\frac{1}{n} \sum_{t=1}^{n} \overleftrightarrow{\mathbf{h}}_t) + \mathbf{b}_d),$$

where $\mathbf{W}_d$ and $\mathbf{b}_d$ are parameters of the transformation, and n is the number of sentences in the document.

The probability of extraction decisions Y conditioned on document X could be factorized as $\Pr(Y|X) = \prod_{t=1}^{n} \Pr(y_t|X, y_{1:t-1})$. The probability of extracting the t-th sentence is computed as

$$\Pr(y_t = 1|X, y_{1:t-1}) = \text{MLP}(\overleftrightarrow{\mathbf{h}}_t, \mathbf{g}_{t-1}, \mathbf{d}), \qquad (1)$$

where $\mathbf{g}_{t-1}$ represents all sentences extracted before time t.MLP$(\cdot)$ means a multilayer perceptron that outputs a probability

$$\begin{aligned}
\text{MLP}(\overleftrightarrow{\mathbf{h}}_t, \mathbf{g}_{t-1}, \mathbf{d}) = \sigma(\tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1 [\overleftrightarrow{\mathbf{h}}_t, \mathbf{g}_{t-1}, \mathbf{d}] \\
+ \mathbf{b}_1) + \mathbf{b}_2)),
\end{aligned}$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$ and $\mathbf{b}_2$ are parameters of MLP and $\sigma(\cdot)$ is the sigmoid function.

Since NES is trained with supervised learning, ground truth extraction labels $(\mathring{y}_1, \cdots, \mathring{y}_n)$ are available during training. Then the representation of sentences selected before or at time t is

$$\mathbf{g}_t = \mathbf{g}_{t-1} + \mathring{y}_t \tanh(\mathbf{W}_g \overleftrightarrow{\mathbf{h}}_t). \qquad (2)$$

The NES model is pretrained by minimizing the negative log-likelihood of the ground truth extraction labels

$$\begin{aligned}
L_{\text{pretrain}}(\Theta) = -\sum_{i=1}^{N} \sum_{t=1}^{N_d} [\mathring{y}_t^i \log \Pr(y_t^i = 1|X_i, \mathring{y}_{1:t-1}^i) \\
+ (1 - \mathring{y}_t^i) \log \Pr(y_t^i = 0|X_i, \mathring{y}_{1:t-1}^i)].
\end{aligned}$$

Reinforced Neural Extractive Summarization Model

After the NES model is trained using supervised learning, it is forced to extract coherent and informative summaries using reinforcement learning with the neural coherence and ROUGE as the reward. In this section, we first introduce the REINFORCE algorithm and then describe our neural coherence model and the ROUGE score reward. The overall training algorithm is illustrated in Algorithm 1.

Reinforcement Learning

The problem of extractive summarization could be formulated as a reinforcement learning problem. The RNES model can be considered as an *agent* that extracts sentences sequentially from documents. At each time step t, the agent is in *state* $s_t = (X, y_{1:t-1})$ which includes the document and previous selections. Agent would take a *action* $y_t \in \{0, 1\}$ that decides whether sentence x_t is extracted or not. After the agent takes the action y_t, it may receive an immediate reward r_t that shows how good the action is. The reward could be delayed. When the agent finishes extracting sentences from the whole document, it also receives an final reward r_{-1} that indicates performance of the entire action sequence $(y_1, y_2, \cdots, y_n)$.

We use the REINFORCE algorithm to train our RNES model. It is a kind of policy gradient method proposed by (Williams 1992), and it maximizes the performance of the

agent by updating its policy. The policy is defined as the probability of taking an action at time t given the state, which is parameterized by Θ:

$$\pi(a|s_t,\Theta) \stackrel{\text{def}}{=} \Pr(y_t = a|s_t,\Theta) \stackrel{\text{def}}{=} \Pr(y_t = a|X, y_{1:t-1},\Theta).$$

In our case, Θ represents all the parameters in the RNES model. We use a shorthand π_Θ to denote the policy π parameterized by Θ. By applying Equation 1, we have

$$\pi_\Theta(a = 1|s_t) = \text{MLP}_\Theta(\overleftrightarrow{\mathbf{h}}_t, \mathbf{g}_{t-1}, \mathbf{d}).$$

Let $s_0 = X$ represents the initial state when no actions are taken yet, and $v_{\pi_\Theta}(s_0)$ be the *value* function that represents the expected *return* starting with state s_0 by following policy π_Θ. Return at time t is defined as $R_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$, where γ is the discount factor. The objective of REINFORCE is defined as maximizing the value of initial state $v_{\pi_\Theta}(s_0)$, or minimizing its negative $L_{RF}(\Theta) = -v_{\pi_\Theta}(s_0)$. Therefore, the parameters should be updated by the gradient of $L_{RF}(\Theta)$ with respect to parameters Θ:

$$\nabla L_{RF}(\Theta) = -\nabla v_{\pi_\Theta}(s_0) = -\sum_{t=1}^{n} \gamma^t \Pr(s_t|s_0,\pi_\Theta) \sum_a q_{\pi_\Theta}(s_t,a) \nabla \pi_\Theta(a|s_t),$$

where $q_{\pi_\Theta}(s,a)$ is the *action-value* function that represents the expected return after taking action a at state s with policy π_Θ.

Since the state space is too large, it is infeasible to compute the exact value of the gradient. We use Monte Carlo sampling to approximate the gradient:

$$\nabla L_{RF}(\Theta) = -\mathbb{E}_{\tilde{y}_t,\tilde{s}_t \sim \pi_\Theta}\left[\gamma^t \tilde{R}_t \nabla \log \pi_\Theta(\tilde{y}_t|\tilde{s}_t)\right], \quad (3)$$

where $\tilde{s}_t$ and $\tilde{y}_t$ are randomly sampled from π_Θ, $\tilde{R}_t$ is the actual return received since $\tilde{s}_t$ and $\tilde{y}_t$. A detailed proof of Equation 3 could be found in (Sutton and Barto 1998) and is omitted for brevity here. The parameters Θ is updated as follows:

$$\Theta \leftarrow \Theta + \gamma^t \tilde{R}_t \nabla \log \pi_\Theta(\tilde{y}_t|\tilde{s}_t). \quad (4)$$

We use $\gamma = 1$ for simplicity in all our settings.

The definition of reward is crucial for reinforcement learning because it determines the optimization direction. To ensure that RNES model extracts coherent and informative summaries, the reward includes both coherence score and ROUGE score, which will be introduced later in this paper. Given a sequence of sampled actions $\tilde{Y} = (\tilde{y}_1, \cdots, \tilde{y}_n)$, the corresponding coherence scores are exploited as immediate rewards $\tilde{r}_t$ and ROUGE as the final reward $\tilde{r}_{-1}$. Therefore, our algorithm is indeed maximizing a weighted sum of coherence and ROUGE score:

$$v_\pi(s_0) \stackrel{\text{def}}{=} \mathbb{E}_{\pi_\Theta}[R_0|s_0] = \mathbb{E}_{\tilde{y}_t,\tilde{s}_t\sim\pi_\Theta}[\tilde{r}_{-1} + \sum_{t=1}^{n} \tilde{r}_t|s_0] = \mathbb{E}_{\pi_\Theta}[\text{ROUGE}(\tilde{G}) + \lambda \text{Coherence}(\tilde{G})|X] \quad (5)$$

Algorithm 1 The overall training algorithm of RNES model. α is the learning rate, χ is a placeholder sentence for bootstrapping the coherence score of the first extracted sentence.

```
Ψ ← train the neural coherence model.
Θ ← pretrain the neural sentence extractor with supervised learning.
loop
    X, H ← sample a document-summary pair from corpus D
    s̃_0 ← X
    Sample an episode s̃_1, ỹ_1, ··· , s̃_n, ỹ_n following π_Θ
    previous ← χ (a placeholder for empty start sentence)
    for each step t = 1 ... n do
        if ỹ_t = 1 then
            r̃_t ← λ Coh_Ψ(previous, x_t)
            previous ← x_t
        else
            r̃_t ← 0
    G̃ ← extract(X, (ỹ_1, ··· , ỹ_n))
    r̃_{-1} = ROUGE(G̃, H)
    for each step t = 1 ... n do
        R̃_t ← Σ_{i=t}^{n} r̃_i + r̃_{-1}
        Θ ← Θ + α R̃_t ∇ log π_Θ(ỹ_t|s̃_t)
```

where $\tilde{G} = \text{extract}(X, \tilde{Y})$ is the sampled summary and λ is the coefficient that balances the two rewards. Coherence$(\tilde{G})$ is the sum of coherence scores of $\tilde{G}$:

$$\text{Coherence}(\tilde{G}) = \sum_{(\tilde{S}_A,\tilde{S}_B)\in adj(\tilde{G})} \text{Coh}(\tilde{S}_A, \tilde{S}_B).$$

The function $\text{Coh}(\cdot,\cdot)$ is defined by the neural coherence model in Equation 6, which will be introduced in the next subsection. Algorithm 1 shows the overall REINFORCE algorithm to train our proposed RNES model.

Neural Coherence Reward

We propose a neural coherence model to compute the cross sentence coherence as the reward of RNES model. This model is built on the ARC-II proposed by (Hu et al. 2014) for sentence matching. The neural coherence model has some advantages over traditional entity grid models. On the one hand, the neural coherence model requires no feature engineering and could be trained in an end-to-end fashion. On the other hand, the neural coherence model uses the distributed text representation which can capture syntactic and semantic coherence pattern by cross-sentence interaction. The architecture of the neural coherence model is shown in Figure 1.

Given two sentences S_A and S_B, In layer 1, it uses sliding windows on both sentences to model all the possible local coherence transition of the two sentences. For segment i on S_A and segment j on S_B, the local coherence transition is computed as

$$\mathbf{z}_{i,j}^{(1)} = g(\hat{\mathbf{z}}_{i,j}^{(0)}) \cdot \text{ReLU}(\mathbf{W}^{(1)}\hat{\mathbf{z}}_{i,j}^{(0)} + \mathbf{b}^{(1)}),$$

where $\mathbf{W}^1$ is the weight parameters for first layer and $\mathbf{b}^1$ is the bias. ReLU is the nonlinear function proposed by (Dahl,

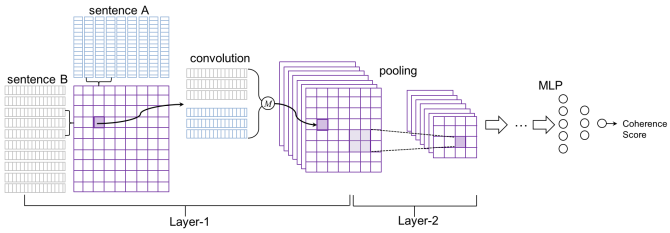

Figure 1: The illustration of neural coherence model which is build on ARC-II proposed by (Hu et al. 2014).

Sainath, and Hinton 2013). $\hat{\mathbf{z}}_{i,j}^{(0)} \in \mathbb{R}^{2k_1 \times D_e}$ is got by concatenating the embeddings of words in S_A and S_B sequentially:

$$\hat{\mathbf{z}}_{i,j}^{(0)} = [e(a_i); ...; e(a_{i+k_1-1}); e(b_j); ...; e(b_{j+k_1-1})],$$

where $g(\hat{\mathbf{z}}_{i,j}^{(0)})$ is the gate function. It equals 0 if all the element of $\hat{\mathbf{z}}_{i,j}^{(0)}$ are 0, and otherwise 1.

Layer 2 takes in the output of layer 1 and performs a max-pooling in each dimension of the cells on non-overlapping 2×2 windows.

$$\mathbf{z}_{i,j}^{(2)} = \max(\mathbf{z}_{2i-1,2j-1}^{(1)}, \mathbf{z}_{2i-1,2j}^{(1)}, \mathbf{z}_{2i,2j-1}^{(1)}, \mathbf{z}_{2i,2j}^{(1)}).$$

Following layer 2, there is more convolution and max-pooling layers, analogous to that of convolutional architecture for image input (LeCun and Bengio 1995). Finally, we obtain the fixed length vector $\mathbf{h}$ and it is fed into a nonlinear transformation with activation function $\tanh$ to compute coherence score of the two sentences:

$$\text{Coh}(S_A, S_B) = \tanh(\mathbf{W}_c \mathbf{h} + \mathbf{b}_c), \tag{6}$$

where $\mathbf{W}_c$ is the weight parameters and $\mathbf{b}_c$ is the bias. Hence, the coherence model will output a coherence score $\text{Coh}(S_A, S_B) \in [-1, 1]$ for any sentence pairs (S_A, S_B).

From the first layer, the neural coherence model can capture the local coherence of two sentences. And it can also obtain higher level coherence representation of S_A and S_B with more convolution and max-pooling layers.

For the training of the neural coherence model, we use a pair-wise training strategy with a large margin objective. Suppose we are given the following triples $(S_A, {S_B}^+, {S_B}^-)$, we adopt the ranking-based loss as objective:

$$L_\Theta(S_A, {S_B}^+, {S_B}^-) = \max(0, 1 + \text{Coh}(S_A, {S_B}^-) - \text{Coh}(S_A, {S_B}^+)).$$

The model is trained by minimizing the above objective, to encourage the model to assign higher coherence score to coherent sentence pairs $(S_A, {S_B}^+)$ than $(S_A, {S_B}^-)$.

ROUGE Score Reward

ROUGE score is used as the final reward to ensure that RNES model extracts reasonably informative sentences. Given a sequence of sampled decisions $(\tilde{y}_1, \cdots, \tilde{y}_n)$, we can get the sequence of extracted sentences $\tilde{G}$. Since the dataset comes with news highlights written by human editors, these manual highlights H is treated as the reference summary. Then ROUGE score $\text{ROUGE}(\tilde{G}, H)$ could be computed and used as a reward for the entire sampled decisions:

$$\tilde{r}_{-1}(\tilde{y}_1, \cdots, \tilde{y}_n) = \text{ROUGE}(\tilde{G}, H).$$

Multiple variants of ROUGE score are proposed by (Lin 2004). Among them, ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) are the most commonly used ones. ROUGE-n (R-n) recall between an extracted summary and a reference summary can by computed as follows:

$$\text{R-}n = \frac{\sum_{s \in \text{reference summary}} \sum_{\text{gram}_n \in s} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{s \in \text{reference summary}} \sum_{\text{gram}_n \in s} \text{Count}(\text{gram}_n)},$$

where n stands for the length of n-gram, $\text{Count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in both the extracted summary and the reference. Similarly we could compute the R-n precision and F1. R-1 and R-2 are special cases of R-n in which $n = 1$ or $n = 2$. R-L is instead computed based on the length of longest common subsequence between the candidate summary and the reference. Since using only variant of ROUGE as reward for training RNES may not increase its performance on other ROUGE variants, we used a combination of ROUGE variants as reward:

$$\text{ROUGE}(G, H) = w_1 \text{R-1}(G, H) + w_2 \text{R-2}(G, H) + w_l \text{R-L}(G, H),$$

where weights w_1, w_2 and w_l are hyperparameters.

The overall training algorithm is illustrated in the Algorithm 1. Since REINFORCE algorithm converges very slowly, we pretrain the RNES model with supervised learning. The neural coherence model is also trained and then fixed for the coherence scoring. During the training of REINFORCE, an sequence of actions and states are sampled according to the policy. Then the coherence model and ROUGE are used for computing the rewards. The parameters of RNES model Θ is then updated according to Equation 4.

Experiments and Results

We used the CNN/Daily Mail dataset originally introduced by (Hermann et al. 2015) to evaluate our RNES model. This dataset contains news documents and highlights crawled from CNN and Daily Mail website, it is commonly used for extractive summarization (Cheng and Lapata 2016; Nallapati, Zhai, and Zhou 2017) and abstractive summarization (Nallapati et al. 2016; See, Liu, and Manning 2017). We used the scripts provided by (Hermann et al. 2015) to download the dataset. It contains 287,226 documents for training, 13,368 documents for validation and 11,490 documents for test. Since the dataset only contains manual summaries and do not have extractive labels, a greedy algorithm similar to (Nallapati, Zhai, and Zhou 2017) is used to generate extraction labels for supervised training of NES.

Results of the Neural Coherence Model

The coherence model needs to be trained before it is used to produce coherence score as the reward in the REINFORCE algorithm. In our experiments, we use 64-dimensional word embeddings which are randomly initialized and finetuned in the process of supervised training. The sizes of all its convolutional kernels are set to be 3. The first convolution layer has 128 filters. The second and third convolution layers contain 256 and 512 filters respectively. Each convolution layer is followed by a max pooling layer performed on the sliding non-overlapping 2×2 windows. The final two fully-connected layers have 512 and 256 hidden units respectively. The maximum sentence length is 50, and sentences longer than this length would be truncated, less will be padded with 0s. The coherence model is trained with stochastic gradient descent (SGD) with batch size 64 and learning rate 0.1.

The training triplets are sampled from the CNN/Daily Mail dataset. The S_A and ${S_B}^+$ are adjacent sentences sampled from the documents, and ${S_B}^-$ is a sentence randomly sampled such that ${S_B}^- \neq {S_B}^+$. To make the task more difficult so that the model finds more fine grained coherence patterns, ${S_B}^-$ is sampled from the same document as $(S_A, {S_B}^+)$ and its distance is less than nine sentences from ${S_B}^+$.

The model is tested on 23K positive pairs sampled from the test set, each accompanied with one negative sample. If the model gives a higher score to the positive sample than the negative sample, it is considered correct. The accuracy is 71.3%, versus 50% accuracy for random guess, which indicates the neural coherence model can capture the cross-sentence coherence.

We also conducted empirical studies on some example outputs of our proposed neural coherence model. Table 1 shows some examples of coherence scoring. The first example shows that the model can exploit co-reference for coherence modeling. The model can capture the semantic coherence between two sentences, such as semantically related words such as "photographer" and "shoot" (example 1), "survey" and "answers" (example 3). Furthermore, the coherence model can discover syntactic patterns such as "As a result ...", and "According to ...", which represents the syntactic coherence across sentences. The third example also shows that there is much noise in our training data. After closer inspection, we found that the ${S_B}^-$ rather than ${S_B}^+$, is the right sentence following S_A. ${S_B}^+$ is the image captions embedded in the article, which should be filtered in the data preprocessing phase. However, thanks to the training on the large-scale text corpus, the neural coherence model can score the right sentence much higher. These examples show that the neural coherence model indeed captures the semantic and syntactic coherence pattern across sentences.

Table 1: Example outputs of neural coherence model.

Sentences	Score
S_A: Terry's career as a **photographer** came after he failed to make it as a punk rock musician.	
${S_B}^+$: **He** got his first big break in 1994 with a **shoot** for Vibe magazine.	**0.9885**
${S_B}^-$: The photographer has also directly music videos in his time.	0.5198
S_A: These days we are increasingly using outdoor space for the occasional barbecue or to relax in a hot tub rather than for tending **flowers**, according to researchers.	
${S_B}^+$: **As a result**, only a handful of traditional **flowers** still grow in English country gardens, with the average one usually containing a mere four species - **daffodils**, **crocuses**, **roses** and **tulips**.	**0.8934**
${S_B}^-$: Sir Roy Strong, the landscape designer and former director of the Victoria and Albert Museum, told the Sunday Times: 'British people used to take pride in having neat gardens with lots of flowers.'	-0.0067
S_A: The same **survey** recently showed that university pupils in Britain have an average of 8.2 sexual partners by the time they reach the middle of their higher education.	
${S_B}^+$: A new survey of university students has revealed that they have had an average of 8.2 sexual partners (picture posed by models)	0.0021
${S_B}^-$: **According to the answers they received**, 22 per cent of students didn't lose their virginity until they were 18 years old, with the second most popular age to have sex for the first time being 16.	**0.9999**

Results of the Reinforced Neural Extractive Summarization Model (RNES)

For the NES model, we use 128-dimensional word embeddings and the vocabulary size is 150,000. The convolution kernels have size 3, 5, 7 with 128, 256, 256 filters respectively. We set the hidden state size of sentence-level GRU to 256, and the document representation size to 512. The MLP has two layers, with 512 and 256 hidden units respectively. We fix the maximum sentence length to 50 and the maximum number of sentences in a document to 80. Sentences or documents that are longer than the maximum length are truncated to fit the length requirement.

The model is trained with stochastic gradient descent (SGD) with batch size 64. When doing supervised training of the NES, ground truth extraction labels are used for computing the cross-entropy loss. The labels are generated from the dataset by greedily selects sentences to maximize its ROUGE similarity compared to manual highlights.

During the training of RNES using reinforcement learning, both the neural coherence model and ROUGE scorer are used to compute the reward. As shown in Equation 5, the hyperparameter λ is used to balance between the two objectives. In our experiments, we explored $\lambda = 1.0, 0.1, 0.01, 0.005$. It is found that when $\lambda = 1.0$ or 0.1, the model favor coherence too much that ROUGE degrades rapidly and the model eventually converges to a policy that selects consecutive sentences that are not informative. However, when $\lambda = 0.005$, the ROUGE objective overpowers coherence, and the coherence rewards drop to approximately zero. We found that $\lambda = 0.01$ is a good trade-off such that both rewards increase and eventually converge.

At test time our models produce summaries using beam

search with beam size 10. To compare with previous works, we adopt the same evaluation metrics as in (Nallapati, Zhai, and Zhou 2017). We use full-length F1 of ROUGE-1, ROUGE-2, and ROUGE-L to evaluate our model. Table 2 shows the performance comparison between our models and baselines. Both of our RNES models (with or without coherence as the reward) outperform current state-of-the-art models and NES by a large margin. The result indicates that the summaries extracted by RNES are of higher quality than summaries produced by previous works.

Table 2: Performance comparison on CNN/Daily Mail test set, evaluated with full-length F1 ROUGE scores (%). All scores of RNES are statistically significant using 95% confidence interval with respect to previous best models.

Model	R-1	R-2	R-L
Lead-3	39.2	15.7	35.5
(Nallapati et al. 2016)	35.4	13.3	32.6
(Nallapati et al. 2017)	39.6	16.2	35.3
(See et al. 2017)	39.53	17.28	35.38
NES	37.75	17.04	33.92
RNES w/o coherence	**41.25**	**18.87**	**37.75**
RNES w/ coherence	40.95	18.63	37.41

Though RNES with the coherence reward achieves higher ROUGE scores than baselines, there is a small gap between its score and that of RNES trained without coherence model. This is because that the coherence objective and ROUGE score do not always agree with each other. Since ROUGE is simply computed based on n-grams and longest common subsequence, it is ignorant of the coherence between sentences. Therefore, enhancing coherence may lead to a drop of ROUGE. However, the 95% confidence intervals of the two RNES models overlap heavily, indicating that their difference in ROUGE is insignificant.

Table 3: Comparison of human evaluation in terms of informativeness(Inf), coherence(Coh) and overall ranking. Lower is better.

Model	Inf	Coh	Overall
RNES w/o coherence	1.183	1.325	1.492
RNES w/ coherence	**1.125**	**1.092**	**1.209**

We also conduct a qualitative evaluation to find out whether the introduction of coherence reward improves the coherence of the output summaries. We randomly sample 50 documents from the test set and ask three volunteers to evaluate the summaries extracted by RNES trained with or without coherence as the reward. They are asked to compare and rank the outputs of two models regarding three aspects: informativeness, coherence and overall quality. The better one will be given rank 1, while the other will be given rank 2 if it is worse. In some cases, if the two outputs are identical or have the same quality, the ranks could be tied, i.e., both of them are given rank 1. Table 3 shows the results of human evaluation. RNES model trained with coherence reward is better than RNES model without coherence reward in all three aspects, especially in the coherence. The result indicates that the introduction of coherence effectively improves the coherence of extracted summaries, and also the overall quality. It is surprising that summaries produced by RNES with coherence are also more informative than RNES without coherence, indicating that ROUGE might not be the gold standard to evaluate informativeness as well.

Table 4 shows a pair of summary produced by RNES with or without coherence. The summary produced by RNES without coherence starts with pronoun 'That' which is referring to a previously mentioned fact, and hence it may lead to confusion. On the other hand, the output of RNES trained with coherence reward includes the sentence "The earthquake disaster ..." before referring to this fact in the second sentence, and therefore is more coherent and readable. This is because the coherence model gives a higher score to the second sentence if it can form a coherent sentence pair with the first sentence. In REINFORCE training, if the second sentence receives a high coherence score, the actions of extracting the first sentence before the second one will be strengthened. The example shows that coherence model is indeed effective in changing the behavior of RNES towards extracting more coherent summary.

Table 4: Examples of extracted summary.

Reference: Peter Spinks from the Sydney Morning Herald reported on Amasia. Within 200 million years, he said the new supercontinent will form. One researcher recently travelled to Nepal to gather further information. He spotted that India, Eurasia and other plates are slowly moving together.
RNES w/o coherence: That's according to one researcher who travelled to the country to study how the Indian and Eurasian plates are moving together. And using new techniques, researchers can now start examining the changes due to take place over the next tens of millions of years like never before. Earth's continents are slowly moving together, and in 50 to 200 million years they are expected to form a new supercontinent called Amasia. In 2012 a study suggested this may be centered on the North Pole. The idea that Earth is set to form a new supercontinent-dubbed Amasia - is not new.
RNES w/ coherence: **The earthquake disaster in Nepal has highlighted how Earth's land masses are already in the process of forming a new supercontinent. That**'s according to one researcher who travelled to the country to study how the Indian and Eurasian plates are moving together. And using new techniques, researchers can now start examining the changes due to take place over the next tens of millions of years like never before. Earth's continents are slowly moving together, and in 50 to 200 million years they are expected to form a new supercontinent called Amasia.

Conclusion

In this paper, we proposed a Reinforced Neural Extractive Summarization model to extract a coherent and informative summary from a single document. Empirical results show that the proposed RNES model can balance between the cross-sentence coherence and importance of sentence efficiently and achieve state of the art performance on the benchmark dataset. In the future work, we will focus on improving the performance of our neural coherence model and introduce human knowledge into the RNES.

References

Ayana; Shen, S.; Liu, Z.; and Sun, M. 2016. Neural headline generation with minimum risk training. *CoRR* abs/1604.01904.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

Barzilay, R., and Lapata, M. 2008. Modeling local coherence: An entity-based approach. *Comput. Linguist.* 34(1):1–34.

Berg-Kirkpatrick, T.; Gillick, D.; and Klein, D. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 481–490.

Cheng, J., and Lapata, M. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Dahl, G. E.; Sainath, T. N.; and Hinton, G. E. 2013. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Proceedings of ICASSP*.

Erkan, G., and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22(1):457–479.

Filippova, K.; Alfonseca, E.; Colmenares, C. A.; Kaiser, L.; and Vinyals, O. 2015. Sentence Compression by Deletion with LSTMs. In *EMNLP*, 360–368.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, 1693–1701.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27*, 2042–2050.

Hu, B.; Chen, Q.; and Zhu, F. 2015. LCSTS: A large scale chinese short text summarization dataset. *CoRR* abs/1506.05865.

LeCun, Y., and Bengio, Y. 1995. Convolutional networks for images, speech and time series. *The Handbook of Brain Theory and Neural Networks* 3361.

Li, J., and Hovy, E. H. 2014. A model of coherence based on distributed sentence representation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *EMNLP*, 2039–2048. ACL.

Lin, H., and Bilmes, J. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 510–520.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., ed., *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 74–81.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Nallapati, R.; Zhou, B.; Gulcehre, C.; and Xiang, B. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th {SIGNLL} Conference on Computational Natural Language Learning*.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 3075–3081.

Nguyen, D. T., and Joty, S. R. 2017. A neural local coherence model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1320–1330.

Nguyen, K.; Boyd-Graber, J.; and Daume, H. I. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Empirical Methods in Natural Language Processing*.

Paulus, R.; Xiong, C.; and Socher, R. 2017. A deep reinforced model for abstractive summarization. *CoRR* abs/1705.04304.

Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *CoRR* abs/1511.06732.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*, 379–389.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *arXiv:1704.04368 [cs]*. arXiv: 1704.04368.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1.

Tang, D.; Qin, B.; and Liu, T. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1422–1432.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yao, J.; Wan, X.; and Xiao, J. 2017. Recent advances in document summarization. *Knowledge and Information Systems* 1–40.

Zhou, X.; Hu, B.; Lin, J.; xiang, Y.; and Wang, X. 2015. Icrc-hit: A deep learning based comment sequence labeling system for answer selection challenge. In *Proceedings of the SemEval 2015*, 210–214.