Neural Headline Generation with Minimum Risk Training

Ayana, Shiqi Shen, Zhiyuan Liu, Maosong Sun

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract

Automatic headline generation is an important research area within text summarization and sentence compression. Recently, neural headline generation models have been proposed to take advantage of well-trained neural networks in learning sentence representations and mapping sequence to sequence. Nevertheless, traditional neural network encoder utilizes maximum likelihood estimation for parameter optimization, which essentially constraints the expected training objective within word level instead of sentence level. Moreover, the performance of model prediction significantly relies on training data distribution. To overcome these drawbacks, we employ minimum risk training strategy in this paper, which directly optimizes model parameters with respect to evaluation metrics and statistically leads to significant improvements for headline generation. Experiment results show that our approach outperforms state-of-the-art systems on both English and Chinese headline generation tasks.

1 Introduction

Automatic text summarization is the process of creating a coherent, informative and short summary for a document. Text summarization approaches can be divided into two typical categories: extractive and generative. Extractive summarization simply selects a subset of existing sentences from original documents as summary. Despite of its simplicity, extractive summarization suffers from some intrinsic drawbacks, e.g., unable to generate coherent and compact summary in arbitrary length, especially when shorter than one sentence.

In contrast, generative summarization builds semantic representations of a document and creates a summary with sentences not explicitly presented in the original document. When the generated summary is required to be a single compact sentence, we name the summarization task as *headline generation* [4].

Generative summarization needs to accurately understand and represent the semantics of original document, and then generate informative summary according to document representations. Most previous works heavily rely on modeling latent linguistic structures of input document, such as syntactic parsing and semantic parsing, which always bring inevitable errors and degrade summarization quality.

Recent years have witnessed great success of deep neural models for various natural language processing tasks [3; 19; 1; 14] including text summarization. Take neural headline generation (NHG) for example, it learns to build a large neural network, which takes a document as input and directly outputs a compact sentence as headline of the document. NHG exhibits the following advantages: (1) NHG is fully data-driven, requiring no human annotation and other linguistic information. (2) NHG is completely end-to-end, which does not explicitly model latent linguistic structures, and thus prevents error propagation. Moreover, the attention mechanism [1] is introduced in NHG, which learns a soft alignment over input document to generate more accurate headline [15].

NHG has achieved great advantages as compared to conventional generative summarization methods. Nevertheless, NHG still confronts an important drawback: current NHG models are mostly optimized with maximum likelihood estimation (MLE) over training data, without taking evaluation metrics of text summarization into consideration. It constraints the expected training objective within word level instead of sentence level and prevents NHG from capturing various aspects of summarization quality.

To address this issue, we argue that it is desirable to build NHG models in which parameters are tuned with respect to specific evaluation criteria such as ROUGE. We apply the *minimum Bayes risk* technique and perform *minimum risk training* (MRT) to develop NHG models. MRT aims at minimizing the expected loss over training data by taking automatic evaluation metric into consideration. Although MRT has been widely used in many NLP tasks such as statistical machine translation [12; 18; 6; 17], it has not been well considered in generative summarization, especially in NHG models.

We conduct experiments on two real-world datasets in English and Chinese respectively. Experiment results show that, NHG with MRT can significantly and consistently improve the summarization performance as compared to conventional NHG models and other baselines. Moreover, we explore the influence of employing different evaluation metrics and find the superiority of our model stable in MRT.

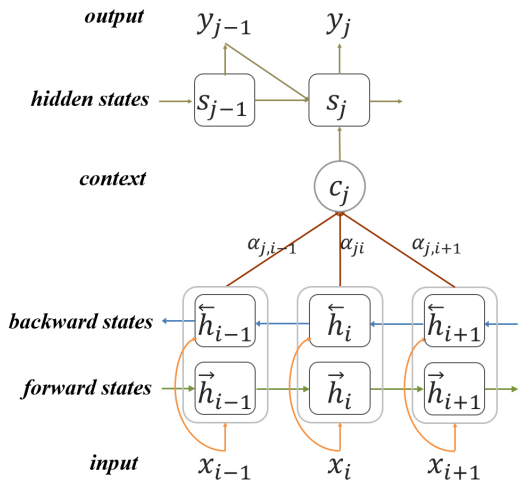

Figure 1: The framework of neural headline generation.

2 Neural Headline Generation Model

We first introduce Neural Headline Generation (NHG) in this section, then introduce minimum risk training (MRT) for NHG in the next section.

We formalize the task of headline generation as follows. Denote the input document $\mathbf{x}$ as a sequence of words $\mathbf{x} = \{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ with M words, in which each word $\mathbf{x}_i$ comes from a fixed word vocabulary V of size $|V|$. Headline generator aims to take $\mathbf{x}$ as input, and generates a short headline $\mathbf{y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ with length $N < M$, so as to maximize the conditional probability of $\mathbf{y}$ given $\mathbf{x}$, i.e., $\arg\max \Pr(\mathbf{y}|\mathbf{x})$.

We follow the Markov assumption to generate headline words $\mathbf{y}_j$ in sequence, and the log conditional probability can be further formalized as:

$$\log \Pr(\mathbf{y}|\mathbf{x};\theta) = \sum_{j=1}^{N} \log \Pr(\mathbf{y}_j|\mathbf{x}, \mathbf{y}_{<j};\theta), \quad (1)$$

where $\mathbf{y}_{<j} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{j-1}\}$, θ indicates model parameters. That is, the j-th word $\mathbf{y}_j$ in headline is generated according to all $\mathbf{y}_{<j}$ generated in past and the input document $\mathbf{x}$. In NHG, we design an encoder-decoder framework to parameterize $\Pr(\mathbf{y}_j|\mathbf{x}, \mathbf{y}_{<j};\theta)$. The framework is shown in Fig. 1.

The encoder of NHG takes the document $\mathbf{x}$ as input and encodes it into a low-dimensional vector $\mathbf{c}$. More specifically, we apply recurrent neural networks (RNN) to get hidden states

$$\mathbf{h}_i = f(\mathbf{x}_i, \mathbf{h}_{i-1}), \quad (2)$$

for each input word $\mathbf{x}_i$, and then we obtain the vector $\mathbf{c} = \mathbf{g}(\{\mathbf{h}_1, \ldots, \mathbf{h}_M\})$, where $\mathbf{g}$ is a function that combines hidden states.

In the decoder of NHG, we also employ RNN to obtain hidden states $\mathbf{s}_j$, and generate headline word by word based on $\mathbf{c}$, $\mathbf{s}_j$ and $\mathbf{y}_{<j}$, i.e.,

$$\Pr(\mathbf{y}_j|\mathbf{x}, \mathbf{y}_{<j}) = \Pr(\mathbf{y}_j|\mathbf{c}, \mathbf{s}_j, \mathbf{y}_{<j}). \quad (3)$$

We introduce the encoder and decoder in detail as follows.

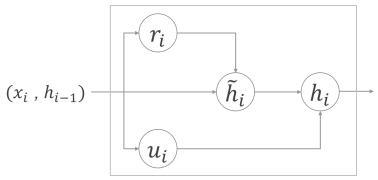

Figure 2: Gated recurrent unit.

2.1 Encoder

in this progress, we apply gated recurrent unit (GRU) to build a bidirectional RNN as the encoder of NHG.

Gated Recurrent Unit

The RNN in NHG is implemented using GRU, which is originally proposed for NMT [3]. As shown in Fig. 2, GRU can adaptively capture dependencies of input sequence by introducing the update gate $\mathbf{u}_i$ and reset gate $\mathbf{r}_i$. The reset gate determines whether to ignore previous hidden states. If the reset gate is close to 1, the update gate controls how much of previous hidden states will pass on. In GRU, $\mathbf{h}_i$ and $\tilde{\mathbf{h}}_i$ are generated hidden state and candidate activation. GRU calculates the i-th hidden state as follows:

$$\mathbf{r}_i = \sigma\big(\mathbf{W}_r\mathbf{E}\mathbf{x}_i + \mathbf{U}_r\mathbf{h}_{i-1}\big) \quad (4)$$

$$\mathbf{u}_i = \sigma\big(\mathbf{W}_u\mathbf{E}\mathbf{x}_i + \mathbf{U}_u\mathbf{h}_{i-1}\big) \quad (5)$$

$$\tilde{\mathbf{h}}_i = \tanh\big(\mathbf{W}_h\mathbf{E}\mathbf{x}_i + \mathbf{U}_h(\mathbf{r}_i \odot \mathbf{h}_{i-1})\big) \quad (6)$$

$$\mathbf{h}_i = \mathbf{u}_i \odot \mathbf{h}_{i-1} + (\mathbf{1} - \mathbf{u}_i) \odot \tilde{\mathbf{h}}_i \quad (7)$$

Here $\sigma()$ is the sigmoid function, $\odot$ indicates element-wise multiplication, $\mathbf{E} \in \mathbb{R}^{D\times V}$ is word embedding matrix, $\mathbf{W}_r, \mathbf{W}_u, \mathbf{W}_h \in \mathbb{R}^{H\times D}$ and $\mathbf{U}_r, \mathbf{U}_u, \mathbf{U}_h \in \mathbb{R}^{H\times H}$ are weighting matrices, with D and H denoting dimensions of word embeddings and hidden states respectively. Note that $\mathbf{h}_0$ is set to $\mathbf{0}$ vector.

Bidirectional RNN

Conventional RNNs typically deal with text sequence from start to end, and build the hidden state of each word only considering its preceding words. It has been verified that, the hidden state should also consider its following words as well. Hence, we apply bidirectional RNN (BRNN) [16] to learn hidden states using both preceding and following words.

As shown in Figure 1, BRNN processes the input document in both forward direction and backward direction with two separate hidden layers calculated with GRUs, obtains the forward hidden states $(\overrightarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_M)$ and the backward hidden states $(\overleftarrow{\mathbf{h}}_1, \ldots, \overleftarrow{\mathbf{h}}_M)$. For each position i, we simply concatenate its both forward and backward states into the final hidden state:

$$\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i, \quad (8)$$

in which $\overrightarrow{\mathbf{h}}_i$ is calculated following Eq. (2) and $\overleftarrow{\mathbf{h}}_i$ is calculated following $\overleftarrow{\mathbf{h}}_i = f(\mathbf{x}_i, \overleftarrow{\mathbf{h}}_{i+1})$, operator $\oplus$ indicates concatenation.

2.2 Decoder

We also use GRU to implement the decoder, which is expected to generate headline words using Eq. (3). The decoder calculates the j-th headline word as follows:

$$\mathbf{r}_j = \sigma\big(\mathbf{W}_r\mathbf{E}\mathbf{y}_{j-1} + \mathbf{U}_r\mathbf{s}_{j-1} + \mathbf{C}_r\mathbf{c}_j\big) \quad (9)$$

$$\mathbf{u}_j = \sigma\big(\mathbf{W}_u\mathbf{E}\mathbf{y}_{j-1} + \mathbf{U}_u\mathbf{s}_{j-1} + \mathbf{C}_u\mathbf{c}_j\big) \quad (10)$$

$$\tilde{\mathbf{s}}_j = \tanh\big(\mathbf{W}_h\mathbf{E}\mathbf{y}_{j-1} + \mathbf{U}_h(\mathbf{r}_j \odot \mathbf{s}_{j-1}) + \mathbf{C}_h\mathbf{c}_j\big) \quad (11)$$

$$\mathbf{s}_j = \mathbf{u}_j \odot \mathbf{s}_{j-1} + (\mathbf{1} - \mathbf{u}_j) \odot \tilde{\mathbf{s}}_j \quad (12)$$

The notations are identical to those of GRUs in the encoder and $\mathbf{C}_r, \mathbf{C}_u, \mathbf{C}_h \in \mathbb{R}^{H\times 2H}$ are weighting matrices as well. We simply set $\mathbf{s}_0 = \tanh(\mathbf{W}_f \overleftarrow{\mathbf{h}}_1)$ with $\mathbf{W}_f \in \mathbb{R}^{H\times H}$.

Also note that, the context vector $\mathbf{c}_j$ conceives attention information of the input document given preterit generated headline words.

$$\mathbf{c}_j = \sum_{i=1}^{M} \alpha_{ji}\mathbf{h}_i, \quad (13)$$

where $\mathbf{h}_i$ is hidden state from the encoder, α_{ji} indicates how much the i-th word $\mathbf{x}_i$ from the original document contributes to generating the j-th word in headline. α_{ji} is computed as follows:

$$\alpha_{ji} = \text{softmax}\big(\mathbf{z}^\top \tanh(\mathbf{W}_\alpha\mathbf{s}_{j-1} + \mathbf{U}_\alpha\mathbf{h}_i)\big), \quad (14)$$

where $\mathbf{z}$ is the weighting vector, and $\mathbf{W}_\alpha$ and $\mathbf{U}_\alpha$ are weighting matrices.

After introducing NHG in details, we describe minimum risk training for NHG in the next section.

3 Minimum Risk Training for NHG

The model parameters of NHG can be estimated with large-scale document-headline pairs. We denote the training set as $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(T)}, \mathbf{y}^{(T)})\}$. Before introducing minimum risk training, we begin with the conventional optimization technique, maximum likelihood estimation, for NHG.

3.1 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) finds optimized parameters which can maximize the log likelihood of generating headlines over training set D:

$$\mathcal{L}_{\text{MLE}}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \log \Pr(\mathbf{y}|\mathbf{x};\theta), \quad (15)$$

where $\Pr(\mathbf{y}|\mathbf{x};\theta)$ is defined in Eq. (1). Therefor, fundamentally the model is trained to maximize $\Pr(\mathbf{y}_j|\mathbf{x},\mathbf{y}_{<j};\theta)$ step by step. By doing so, the training procedure will inevitably lose global information. On the other hand, during training, $\mathbf{y}_{<j}$ are authentic words from standard headline. However during testing, $\mathbf{y}_{<j}$ are predicted by model and may not be correct, which lead to error propagation and inaccurate headline generation. Minimum risk training on the other hand, can improve these problems.

3.2 Minimum Risk Training

Minimum risk training (MRT) aims to minimize the expected loss, i.e. risk, over the training data.

Given a document $\mathbf{x}$, we define $\mathcal{Y}(\mathbf{x};\theta)$ as the set of all possible headlines generated by NHG with parameters θ. Regarding $\mathbf{y}$ as the gold-standard headline of $\mathbf{x}$, we define the loss of a generated headline $\mathbf{y}'$ as semantic distance between $\mathbf{y}'$ and the standard $\mathbf{y}$, denoted as $\Delta(\mathbf{y}',\mathbf{y})$. MRT defines the objective function as follows:

$$\mathcal{L}_{\text{MRT}}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \mathbb{E}_{\mathcal{Y}(\mathbf{x};\theta)}\Delta(\mathbf{y}',\mathbf{y}). \quad (16)$$

Here $\mathbb{E}_{\mathcal{Y}(\mathbf{x};\theta)}$ indicates the expectation over the set $\mathcal{Y}(\mathbf{x};\theta)$. Thus the objective function of MRT can be further formalized as:

$$\mathcal{L}_{\text{MRT}}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\mathcal{Y}(\mathbf{x};\theta)} \Pr(\mathbf{y}'|\mathbf{x};\theta)\Delta(\mathbf{y}',\mathbf{y}). \quad (17)$$

Moreover, it is usually time-consuming and inefficient to enumerate all possible instances in $\mathcal{Y}(\mathbf{x};\theta)$. For simplicity, we only sample a significant subset $\mathcal{S}(\mathbf{x};\theta) \subset \mathcal{Y}(\mathbf{x};\theta)$ to approximate the probability distribution, and formalize the objective function as:

$$\mathcal{L}_{\text{MRT}}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\mathcal{S}(\mathbf{x};\theta)} \frac{\Pr(\mathbf{y}'|\mathbf{x};\theta)^\epsilon}{\sum_{\mathbf{y}^*\in\mathcal{S}(\mathbf{x};\theta)}\Pr(\mathbf{y}^*|\mathbf{x};\theta)^\epsilon}\Delta(\mathbf{y}',\mathbf{y}), \quad (18)$$

in which ϵ is a hyper-parameter [12] that controls the smoothness of the objective function. Selection of this parameter directly affects the optimization results.

MRT exploits $\Delta(\mathbf{y}',\mathbf{y})$ to measure the loss, which enables us to optimize NHG models with respect to the specific evaluation metrics of the task. As we know, the most widely adopted evaluation metric for document summarization is ROUGE [11] (Recall-Oriented Understudy of Gisting Evaluation). Hence, we simply measure the semantic distance $\Delta(\mathbf{y}',\mathbf{y})$ with ROUGE.

3.3 ROUGE

ROUGE automatically measures summary quality by comparing computer-generated summaries to standard summaries created by human. ROUGE is the common evaluation metric in Document Understanding Conference (DUC), a large-scale summarization evaluation sponsored by NIST [11]. The basic idea of ROUGE is to count the number of overlapping units such as overlapped n-grams, word sequences, and word pairs between computer-generated summaries and the standard summaries.

In this project, we consider two types of ROUGE: ROUGE-N and ROUGE-L. ROUGE-N counts n-grams, and ROUGE-L counts longest common sub-sequences.

Suppose $\mathbf{y}'$ is the generated summary, and $\mathbf{y}$ is the standard summary. ROUGE-N is defined as follows:

$$\text{ROUGE-N} = \frac{\sum_{\text{gram}_\text{N}\in\mathbf{y}} M(\text{gram}_\text{N})}{\sum_{\text{gram}_\text{N}\in\mathbf{y}} C(\text{gram}_\text{N})}, \quad (19)$$

where N indicates the type of N-gram (e.g., uni-gram and bi-gram, corresponding to ROUGE-1 and ROUGE-2),

$M(\text{gram}_\text{N})$ is the number of n-grams matched between $\mathbf{y}'$ and $\mathbf{y}$, and $C(\text{gram}_\text{N})$ is the total number of n-grams in $\mathbf{y}$.

ROUGE-L is formalized as:

$$\text{ROUGE-L} = \frac{(1+\beta^2)R_\text{L}P_\text{L}}{R_\text{L}+\beta^2 P_\text{L}}. \quad (20)$$

Here β is the harmonic factor between recall R_L and precision P_L, which are defined as:

$$R_\text{L} = \frac{\text{Lcs}(\mathbf{y}', \mathbf{y})}{\text{Len}(\mathbf{y})}, \quad P_\text{L} = \frac{\text{Lcs}(\mathbf{y}', \mathbf{y})}{\text{Len}(\mathbf{y}')}, \quad (21)$$

where $\text{Lcs}(\mathbf{y}', \mathbf{y})$ is the length of longest common subsequence between $\mathbf{y}'$ and $\mathbf{y}$, and $\text{Len}(\mathbf{y})$ is the length of $\mathbf{y}$.

Fundamentally ROUGE is a similarity metric. Hence, we simply define $\Delta(\mathbf{y}', \mathbf{y}) = -\text{ROUGE-}\{1,2,\text{L}\}$ in MRT.

4 Experiments

We conduct experiments on both English and Chinese datasets and compare the performance of our model with several baseline systems. In this section we introduce datasets, baseline systems and experiment results in details.

4.1 Datasets

English Dataset

For English, we utilize the English Gigaword Fifth Edition [13] for training models. It is one of the largest static corpus of English news, consisting of nearly 10 million news articles from 7 news outlets, with a total of more than 4 billion words. In order to compare our work with [15], we take the same preprocessing techniques as theirs.

We opt the first sentence of each news article and pair it with its corresponding headline as an article-headline pair. To avoid noises in articles and headlines that may influence the performance, we filter out bylines, extraneous editing marks and question marks. The training set contains approximately 4 million article-headline pairs after the filtering step. We take DUC-2003 evaluation dataset as our validation set to determine the hyper-parameters resulting the best performance in minimum risk training. DUC-2003 contains 624 short articles, each of which corresponds to four human-generated reference headlines.

We use the dataset from DUC-2004 Task-1 as our test set. The dataset consists of 500 news articles from Associated Press Wire services and New York Times. Each article is paired with four human-generated reference headlines, with maximum length of 75 characters. [1]

Chinese Dataset

We also implement experiments on a Chinese dataset LCSTS [8]. This dataset consists of article-headline pairs extracted from Sina Weibo [2], a Chinese social media that allows users disseminate and share information with their friends. A typical news article posted in Weibo is limited to 140 Chinese characters, and the corresponding headline is usually set in a pair of square brackets at the beginning of the news article.

[1]The dataset can be downloaded from `http://duc.nist.gov/` with agreements.

[2]The website of Sina Weibo is `http://weibo.com/`

Table 1: Hyper-parameters used in NHG.

	English NHG	Chinese NHG
word embedding size	620	400
hidden unit size	1,000	500
vocabulary size	30,000	3,500

The LCSTS consists of three parts. Part-I incorporates about 2.4 million article-headline pairs. Part-II and Part-III contain article-headline pairs with human-labeled scores, indicating the relatedness between the article and its headline. The higher score, the more related they are to each other. Part-II consists of 10,666 and Part-III only 1,106 article-headline pairs. Each pair in Part-II is labeled by only one annotator, and in Part-III is by three annotators.

We utilize Part-I of LCSTS for model training, and we opt the article-title pairs in Part-III with scores greater than or equal to 3 for testing. It is worth mentioning that, we take Chinese characters as inputs of NHG instead of words, in order to prevent the errors in Chinese word segmentation.

4.2 Baseline Systems

In this progress, we compare our model against the following baselines on English headline generation task:

- **TOPIARY** [21] is the winner system of DUC2004 Task-1. This system utilizes linguistic-based sentence compression method and unsupervised topic detection at the same time, and achieves good performance.
- **MOSES+** [15] generate headlines based on a widely-used phrase-based machine translation system MOSES [10]. The MOSES+ system further takes two steps to improve the quality of generated headlines. One step is to enlarge phrase table and the other is to use MERT to tune models.
- **ABS** and **ABS+** [15] are attention-based neural models to generate short summaries given a sentence. The difference between the two systems is that, ABS+ takes an extractive tuning procedure to revise model parameters based on ABS. They both utilize DUC-2003 data as validation set to take a MERT tuning step.

Among them, **MOSES+**, **ABS** and **ABS+** take the same dataset for learning.

4.3 Implementation Details

In MRT, we initialize model parameters using the optimized parameters learned from NHG with MLE. In particular, the size of $\mathcal{S}(\mathbf{x};\theta)$ for each $\mathbf{x}$ is important: when the size is too small, the sampling will not be sufficient and hurt the performance; when the size is too large, the learning time will grow correspondingly. In this progeress, we set the size to 100 to achieve a trade-off between effectiveness and efficiency. We select top-ranked summaries, i.e., those with the largest generation probabilities, generated by the up-to-date NHG model into $\mathcal{S}(\mathbf{x};\theta)$. The hyper-parameter ϵ in Eq.(18) is set to 5×10^{-3}.

Table 1 shows most of the hyper-parameters that we use in our systems and they remain the same in both MLE training

and MRT training and we have not tuned them. We keep a fix-sized vocabulary for training and evaluation.

Those words that are not included in this vocabulary are mapped to a special token [UNK]. When generating headlines, we utilize a [UNK] replacement technique following the idea in [9]. For English dataset, we preprocess with tokenization, lower-casing. For both English and Chinese, we also replace all digits with #.

All models are trained on Tesla K20 GPU. For NHG+MLE on the English dataset, it takes about 3.75 hours to conduct $10,000$ of iterations, and for NHG+MRT, it is about 5.5 hours. The training time of MRT is a bit longer than MLE because MRT has to learn on 100 sampled headlines for each article.

4.4 Experiment Results and Analysis

We utilize ROUGE [11] that introduced in Section 3.3 to measure the performance of various models. For each model, we report ROUGE-1, ROUGE-2 and ROUGE-L scores in the form of percentage.

Evaluation Results on English and Chinese

Table 2 shows the evaluation results of English headline generation on DUC-2004. TOPIARY, MOSES+, ABS and ABS+ are baselines introduced in Section 4.2. Since we carry out training and evaluation on the same dataset, and hence we simply use the evaluation results reported in [15]. NHG+MLE and NHG+MRT are neural headline generation models we described in Section 2 learned with MLE and MRT respectively.

Table 2: Experiment results on English DUC-2004.

	ROUGE-1	ROUGE-2	ROUGE-L
TOPIARY	25.12	6.46	20.12
MOSES+	26.50	8.13	22.85
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
NHG+MLE	26.78	9.17	23.91
NHG+MRT	**29.34**	**10.67**	**26.10**

From Table 2 we can observe that: (1) NHG learned with MLE achieves competitive performance to the best baseline systems, especially on ROUGE-2 and ROUGE-L, and is slightly worse than ABS+ on ROUGE-1. This indicates that NHG is effective for headline generation. (2) NHG learned with MRT significantly and consistently outperforms NHG with MLE under three evaluation metrics.

The similar results are observed from the Chinese dataset as well, as shown in Table 3. NHG with MRT improves ROUGE scores up to nearly 3 points to NHG with MLE.

The significant superiority on both English and Chinese suggests that, MRT is more effective for learning NHG models as compared to MLE.

Effectiveness of Evaluation Metrics

We are interested in the effectiveness of utilizing various evaluation metrics in MRT, i.e., ROUGE-1, ROUGE-2 and ROUGE-L, for headline generation performance.

Table 3: Experiment results on Chinese LCSTS.

	ROUGE-1	ROUGE-2	ROUGE-L
NHG+MLE	34.92	23.31	32.74
NHG+MRT	**37.87**	**25.43**	**35.33**

Table 4 and 5 show the results of MRT with various evaluation metrics on English validation and test set respectively.

Table 4: Experiment results on English validation set.

	ROUGE-1	ROUGE-2	ROUGE-L
NHG+MLE	24.24	7.75	21.96
NHG+MRT (R-1)	**27.03**	9.28	**24.44**
NHG+MRT (R-2)	26.73	**9.47**	24.23
NHG+MRT (R-L)	26.61	9.20	24.37

Table 5: Experiment results on English test set.

	ROUGE-1	ROUGE-2	ROUGE-L
NHG+MLE	26.78	9.17	23.91
NHG+MRT (R-1)	**29.34**	9.79	**26.10**
NHG+MRT (R-2)	28.86	**10.67**	25.93
NHG+MRT (R-L)	28.62	10.14	25.97

From Table 4 and 5, we observe that: (1) All NHG+MRT models with three evaluation metrics consistently outperform NHG+MLE. This indicates that the MRT technique is robust when evaluation metric varies. (2) It is straightforward that, NHG+MRT models with ROUGE-1 and ROUGE-2 perform slightly better when evaluated using the corresponding metrics. But when evaluated using ROUGE-L, NHG+MRT with ROUGE-L does not perform the best. The reason may be that, MRT with ROUGE-1 and ROUGE-2 is more compatible with the RNN encoder-decoder architecture. In contrast, ROUGE-L measures similarity based on longest common subsequences, which cannot be well considered in RNN.

Case Study

To demonstrate the effectivenesses of MRT, we present several example outputs for comparison as shown in Table 6. From these examples we can observe that: (1) NHG with MRT generally is able to capture the important part of a document. For example in Article 1, the main subject is Honduras prepared for hurricane. NHG+MRT can successfully find the theme and generate a headline about Honduras, but NHG+MLE failed. There is a similar situation in Article 2. (2) When both systems capture the correct subject, NHG+MRT can grasp and generate more informative headline. As shown in Article 3, NHG+MLE missed the information who reported the fact, but NHG+MRT provided concisely with "U.N.:". (3) NHG+MLE usually suffers from generating repeated words or phrases as we present in Article 4. NHG+MLE repeats the phrase "*Asian games*" twice and leads to a semantically incomplete headline. NHG+MRT seems to be able to overcome this issue, benefitting from directly optimizing sentence-level ROUGE.

In summary, all these examples indicate the effectiveness

Table 6: Examples on English dataset.

Article (1):	Honduras braced for potential catastrophe Tuesday as hurricane Mitch roared through the northwest caribbean, churning up high waves and intense rain that sent coastal residents scurrying for safer ground.
Standard:	Honduras , other caribbean countries brace for the wrath of hurricane Mitch
NHG+MLE:	Hurricane Mitch roars through northwest caribbean
NHG+MRT:	Honduras braces for hurricane Mitch
Article (2):	Making their first collective decision about monetary policy , the ## european nations launching a common currency on jan . # cut interest rates Thursday in a surprise move that won market confidence .
Standard:	Eleven european nations cut interest , win market confidence in euro
NHG+MLE:	Euro nations launch common currency
NHG+MRT:	Euro nations cut interest rates
Article (3):	Years of food shortages have stunted the growth of millions of north Korean children, with two-thirds of children under age seven suffering malnourishment, U.N. experts said Wednesday
Standard:	Food shortage stunt north Korean children.
NHG+MLE:	Food shortages hit north Korea
NHG+MRT:	U.N. : food shortages stunted growth of north Korean children
Article(4):	Saudi Arabia's abrupt withdrawal from the asian games left organizers scrambling Thursday to change schedules and thai diplomats mulling a decade of relations strained by jewel theft and the murder of diplomats.
Standard:	Saudi Arabia leaves Asian games ; Thai theft of Saudi jewels may be reason
NHG+MLE:	Asian games pull out of Asian games
NHG+MRT:	Saudi Arabia withdraws from Asian games

of neural headline generation with minimum risk training and its superiority over maximum likelihood estimation.

5 Related Work

Headline generation is a well-defined task standardized in DUC-2003 and DUC-2004. Various approaches have been proposed for headline generation: rule-based, statistical-based and neural-based, introduced in detail as follows.

The rule-based models create a headline for a news article using handcrafted and linguistically motivated rules to guide the choice of a potential headline. Hedge Trimmer [4] is a representative example of this approach which creates a headline by removing constituents from the parse tree of the first sentence until it reaches a specific length limit. Systems in this approach are in accordance with human intuition and easy to understand. However, it is unrealistic and impossible to induce every single rule due to the complexity of human languages.

The statistical-based methods make use of large scale training data to learn correlations between words in headlines and those in articles. For example, [2] applies statistical models for content selection and surface realization to produce headlines. The best system on DUC-2004, TOPIARY [21] combines both linguistic and statistical information to generate headlines. There are also methods make use of knowledge bases to generate better headlines. For example, [20] utilizes Wikipedia to extract features like word inlinks, outlinks and categories to select keywords from an article and constitute a headline.

Recently, with the advances of deep neural networks, there are growing works that design neural networks for headline generation. [15] proposes an attention-based model to generate headlines. This model introduces attention mechanism into sentence compression without recurrent unit. [5] proposes a recurrent neural network with long short term memory (LSTM) [7] for headline generation. This model regards headline generation as sentence compression, with words annotated as preserved or not to form the compressed sentence. This model is essentially an extractive model, unable to generate headlines with those words out of original document, which to some extent restricts its applicability.

In this work, we propose the NHG model realized by a bidirectional recurrent neural network with gated recurrent units. We also propose to apply minimum risk training (MRT) to optimize parameters of NHG model. MRT has been widely used in machine translation [12; 18; 6; 17], but less been explored in document summarization. To the best of our knowledge, this work is the first attempt to utilize MRT in neural headline generation.

6 Conclusion and Future Work

In this progress, we build an end-to-end neural headline generation model, which does not require heavy linguistic analysis and is fully data-driven. We apply minimum risk training for model learning, which is able to take specific evaluation metrics into consideration. Evaluation results show significant and consistent improvements of NHG with MRT over both English and Chinese datasets, as compared to other baselines including NHG with MLE.

There are still many open problems to be explored as future work: (1) Besides article-headline pairs, there are also rich plain text data not considered in NHG training. We will investigate the probability of integrating these plain text to enhance NHG for semi-supervised learning. (2) We will investigate the hybrid approach of incorporating NHG with other successful headline generation approaches like sentence compression models. (3) Both input and output of NHG are typically in sentence level. We will investigate the effectiveness of neural models and MRT on more complicated summarization tasks like single document summarization and multiple document summarization.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015.

[2] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. Headline generation based on statistical translation. In *Proceedings of ACL*, pages 318–325, 2000.

[3] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. pages 1724–1734, 2014.

[4] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of HLT-NAACL*, pages 1–8, 2003.

[5] Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. Sentence compression by deletion with lstms. In *Proceedings of EMNLP*, pages 360–368, 2015.

[6] Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. Learning continuous phrase representations for translation modeling. In *Proceedings of ACL*, pages 699–709, 2014.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, pages 1735–1780, 1997.

[8] Baotian Hu, Qingcai Chen, and Fangze Zhu. Lcsts: A large scale chinese short text summarization dataset. In *Proceedings of EMNLP*, pages 1967–1972, 2015.

[9] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of ACL-IJCNLP*, pages 1–10, 2015.

[10] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180, 2007.

[11] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL*, pages 74–81, 2004.

[12] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167, 2003.

[13] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword fifth edition, june. 2011.

[14] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[15] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*, pages 379–389, 2015.

[16] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, pages 2673–2681, 1997.

[17] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*, 2015.

[18] David A Smith and Jason Eisner. Minimum risk annealing for training log-linear models. In *Proceedings of COLING/ACL*, pages 787–794, 2006.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112, 2014.

[20] Songhua Xu, Shaohui Yang, and Francis Chi-Moon Lau. Keyword extraction and headline generation using novel word features. In *Proceedings of AAAI*, 2010.

[21] David Zajic, Bonnie Dorr, and Richard Schwartz. Bbn/umd at duc-2004: Topiary. In *Proceedings of HLT-NAACL*, pages 112–119, 2004.