Conceptual Multi-Layer Neural Network Model for Headline Generation

Yidi Guo*[1,3], Heyan Huang[1,2], Yang Gao[1,2], and Chi Lu[1,3]

1.Beijing Institute of TechnologyBeijing, China
2.Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications
3.Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing 100048, P.R.China
{gyd409274478,hhy63,gyang,luchi}@bit.edu.cn

Abstract. Neural attention-based models have been widely used recently in headline generation by mapping source document to target headline. However, the traditional neural headline generation models utilize the first sentence of the document as the training input while ignoring the impact of the document concept information on headline generation. In this work, A new neural attention-based model called concept sensitive neural headline model is proposed, which connects the concept information of the document to input text for headline generation and achieves satisfactory results. Besides, we use a multi-layer Bi-LSTM in encoder instead of single layer. Experiments have shown that our model outperforms state-of-the-art systems on DUC-2004 and Gigaword test sets.

Keywords: Attention-based, Concept, Multi-layer Bi-LSTM

1 Introduction

Text summarization is the task of generating a short summary or headline that captures the subject content of a document. It is expected to understand the core meaning of the documents and then produce a coherent, informative but brief summarization of the source document. And We name the summarization task as headline generation [1] when the generated summary required to be a single sentence.

The main approaches of text summarization can be divided into two categories: extractive and generative. Most extractive summarization systems extract parts of the document (words or sentences) that are deemed interesting by some metric and joint them to form a summary. Despite of its simplicity, the summary always is awkward or grammatically strange. In contrast, generative summarization is to simply summarize as humans do. It aims at comprehending a document and generating a coherent and concise summary, which using some vocabulary unseen in the source document.

Recently, sequence-to-sequence (seq2seq) model [2, 3], which maps a sequential input into another sequential output, has been successfully applied on various natural language processing tasks. Especially, the seq2seq model with attention mechanism [4]

* Corresponding author.

has achieved overwhelming successes in machine translation (MT) and speech recognition. Since the superiority of the seq2seq model in sequential language generation that breaks the traditional gaps NLP area, it has also been successfully applied in text summarization [5, 8, 18] and creatively used in headline generation [14, 17].

Despite of the identical objective of language generation for MT and our targeted headline generation task, there are still some differences. MT is a process of language generation, which has a strong one-to-one word-level alignment between input (source) and output (translation), and the length of output is typically close to the length of the input. But the output (headline) of headline generation is typically very short and does not depend on the length of the input. Headline generation is a process of information compression, which uses a lossy manner to compress the source document and preserve the key information. Therefore, The quality of information compression will directly affect the results of the final headline generation. Additionally, compare to the MT, headline generation is more subjective to what people are interested in. As such, a suitable headline should always adjust to some specific concepts. Specifically, the different concepts of news tend to use different words, fields and styles, which inspired us to use the concept information to play a guiding role in headline generation.

To integrate the concept of news for headline generation, in this paper, we propose a novel neural network framework that completes information compression and coherent language generation, specific to the concept sensitive headline generation. In this new model, called concept sensitive neural headline generation model (CNHG), a multi-layer Bi-LSTM encoder is extended by seamlessly adding document-based conceptual information construction model, which can guide the generation model in a manner of latent concept way. More specifically, in the encoding process, the key features was extracted from each document by TextRank algorithm, and then correspondingly maps to several regularized concepts by using probabilistic-based Probase knowledge database. In this way, the document-based concepts were extracted and embedded into the source information encoders. Empirical experiment results show that our model beats the state-of-the-art baseline on multiple English data sets.

The main contributions of this paper include: (i) The core of information compression and coherent language generation for headline generation is validated and accomplished in terms of using multi-layer of Bi-LSTM encoder of seq2seq attention model. (ii) The conceptual information is creatively incorporated into the encoder of the neural headline generation model as a latent guidance for generating more focused and desired headlines. (iii) We conducted extensive experiments to compare the effectiveness of our proposed CNHG model with other state-of-the-art models in benchmark datasets of DUC2004 and Gigaword.

2 Related Work

The task of headline generation was standardized around the DUC-2003 and DUC-2004 competitions [10]. Most of the work is focused on the extractive summarization before the development of the neural network, there has been little research on the generative summarization. The TOPIARY system [11] combines both linguistic and statistical information, which performed best in DUC-2004 Task-1. Later, syntactic and

semantic features were used in headline generation [12, 13]. And MOSES, a widely-used phrase-based machine translation system [16], was directly used as a method to generate headline which named MOSES+ by [5].

Recently, the seq2seq neural model has been applied successfully to various natural language processing tasks. For Headline generation task, Neural Headline Generation (NHG) model has been used widely. Rush et al. [5] proposed the method which combines a feed-forward neural language model with an attention-based encoder have shown significant performance. Chopra et al. [8] extended the model by [5] with a recurrent neural network and the attention mechanism to improve the performance. Method [14] incorporated the structural syntactic and semantic information in a baseline neural attention-based model to generate headline. Based on NHG model, more tricks or methods were added to improve the performance. Model [18] paid attention to vocabulary size and implemented a trick that constructs the vocabulary of documents in each mini-batch respectively. Besides, Gulcehre et al. [15] proposed a method to deal with the rare and unseen words in natural language generation.

However, to our knowledge, not any research has devoted to discover the effect of concept in NHG model. Therefore, in this work, we propose the CNHG model, implemented by a bidirectional recurrent neural network with a multi-layer Bi-LSTM encoder and also encoded by the concept information from source documents. The concept information plays an important role in training and guides the model learn the desired content and concise expression in saliency.

3 Model

Neural Headline Generation aims at generating a brief sentence from the source document. The framework of our proposed CNHG model is a multi-layer Bi-LSTM encoder and a mono-layer of LSTM decoder. Based on the model, conceptual embedding vectors are fed into the encoder to guide the headline generation process.

3.1 Multi-Layer Encoder NHG

Compared to the traditional NHG model, we used a multi-layer Bi-LSTM encoder which means the output of previous encoder layer is the input of next encoder layer. The model we named 4-NHG includes a 4 layer Bi-LSTM encoder and a mono-layer LSTM decoder with the attention mechanism. Fig. 1 shows the framework of the multi-layer Bi-LSTM encoder model.

Long Short-Term Memory Networks Long Short Term Memory (LSTM) was first proposed in [6] to address the issue that standard recurrent neural networks (RNNs) are unable to learn the long-term dependencies. LSTM is a special kind of recurrent neural networks, which are explicitly designed to avoid the long-term dependency problem.

The architecture of LSTM is controlled by a set of gates when it processes an input sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_{|\mathbf{x}|}\}$ and generates a sequence of output states $\mathbf{h} = \{h_1, h_2, \ldots, h_{|\mathbf{x}|}\}$. The input gate i_t to control which part of new information would be used in current memory cell, the forget gate f_t to control what information would be

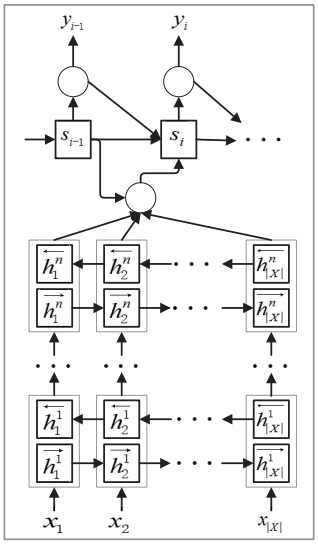

Fig. 1. The framework of the multi-layer Bi-LSTM encoder model

throw away from the old memory cell, and the output gate o_t to control which part of the cell state would be output based on the memory cell. All those gates are combined to decide how to update the current memory cell c_t and the current hidden state h_t. At each time step, LSTM takes the input text x_t, previous hidden state h_{t-1}, previous cell c_{t-1} as input and generates c_t, h_t based on the following formulas:

$$\begin{aligned} f_t &= \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{c_t} &= tanh(\mathbf{W}_c \cdot [h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(\mathbf{W}_o \cdot [h_{t-1}, x_t] + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{1}$$

Here, σ and $tanh$ are activate functions which refer to the logistic sigmoid function and the hyperbolic tangent function. The symbols $\odot$ is an operation which denotes the element-wise multiplication.

Encoder-Decoder The multi-layer Bi-LSTM encoder model encodes input text $\mathbf{x}$ into a sequence of hidden states $\{h_1, \ldots, h_{|\mathbf{x}|}\}$, and uses the attention mechanism to get the context vector c_i for decoder time step i based on $\{h_1, \ldots, h_{|\mathbf{x}|}\}$. The model generates the headline $\mathbf{y}$ by a decoder is based on the context vector c_i. Following a Markov process, the probability over the headline $\mathbf{y}$ is modeled by a product of individual conditional probabilities with parameters θ:

$$P(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^{|\mathbf{y}|} p(y_t|\{y_1, \ldots, y_{t-1}\}, \mathbf{x}, \theta) \tag{2}$$

Decoder The decoder is a mono-layer LSTM which is trained to predict the next output word y_i given the context vector c_i and decoder hidden state s_i for time step i. The above conditional probability is defined as

$$p(y_i|\{y_1,\ldots,y_{i-1}\},\mathbf{x},\theta) = g(s_i,c_i) \tag{3}$$

where s_i is computed by

$$s_i = g(s_{i-1},y_{i-1},c_i) \tag{4}$$

The context vector c_i depends on the output of the encoder module $\{h_1,h_2,\ldots,h_{|\mathbf{x}|}\}$. We computed the context vector c_i as

$$\begin{aligned} u_t^i &= f(s_{i-1},h_t) \\ a_t^i &= softmax(u_t^i) \\ c_i &= \prod_{t=1}^{|\mathbf{x}|} a_t^i h_t \end{aligned} \tag{5}$$

Encoder The encoder is a multi-layer bidirectional LSTM. With input text $\mathbf{x} = \{x_1,x_2,\ldots,x_{|\mathbf{x}|}\}$, the multi-layer forward LSTM calculates a sequence of *forward hidden states* $\mathbf{h}^{(f)} = (h_1^{(f)},\ldots,h_{|\mathbf{x}|}^{(f)})$ while the multi-layer backward LSTM calculates a sequence of *backward hidden states* $\mathbf{h}^{(b)} = (h_1^{(b)},\ldots,h_{|\mathbf{x}|}^{(b)})$. Then $\mathbf{h}^{(f)}$ and $\mathbf{h}^{(b)}$ are concatenated to get $\mathbf{h}$ as

$$\begin{aligned} h_t &= h_t^{(f)} \oplus h_t^{(b)} \\ \mathbf{h} &= \{h_1,h_2,\ldots,h_{|\mathbf{x}|}\} \end{aligned} \tag{6}$$

3.2 Concept Sensitive NHG

Each input text has a thematic concept information, we want to use this thematic concept information to provide a guiding role in the training process. Therefore, we combined the concept information with the above multi-layer encoder model to get a new model, namely Concept Sensitive NHG model. It includes two steps: obtaining concept information and training a concept sensitive model. The probability is defined as

$$P(\mathbf{y}|\mathbf{x},\theta,\theta_c) = \prod_{t=1}^{|\mathbf{y}|} p(y_t|\{y_1,\ldots,y_{t-1}\},\mathbf{x},\theta,\theta_c) \tag{7}$$

where θ_c is the concept information of the input text $\mathbf{x}$, and the θ_c can be formed by the following concept generation process.

Concept Generation Concept generation from a document can be conducted as two steps, which are keyword extraction and word-to-concept mapping. In this paper, TexRank is used to extract a list of keywords, and probabilistic-based Probase knowledge database[1] is used to map keywords to key concepts.

[1] It can be downloaded from https://concept.msra.cn

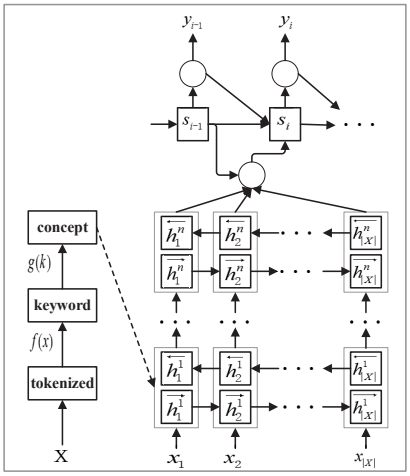

Fig. 2. The framework of Concept Sensitive NHG model with a multi-layer Bi-LSTM encoder

TextRank algorithm is a graph-based sorting algorithm for text. The basic idea of TextRank comes from Google's PageRank algorithm, which divides the text into several units (words and sentences) to establish the graph model and uses the voting mechanism to sort the important components in the text. TextRank algorithm only uses the information of a single document itself to implement keyword extraction, which greatly reduces the extraction time. We first construct a directed weighted graph $G = (V, E)$ for the input document and then the set of keywords U is computed as

$$\begin{aligned} WS(V_i) &= (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \\ U &= Top_n(WS(V_1), \ldots, WS(V_{|V|})) \end{aligned} \tag{8}$$

Here, d is the damping factor, $In(V_i)$ is a set of points pointing to V_i, $Out(V_i)$ is a set of points that V_i points to and $WS(V_i)$ is the score of V_i. The symbols Top_n is a function to obtain the highest score of n keywords, in there, we set up n as 5.

After obtaining the set of keywords, we use Probase to gain the concept information. The core version of IsA data in Probase was mined from billions of web pages, and this data contains more than 5.4 million unique concepts, 12.3 million unique instances. We took the $1,000$ most frequently occurring concepts as a regularized set of concepts.

Probase calculates the concept based on the concept distribution of $p(c|w)$, which is a word-to-concept mapping. The concept information θ_c is computed follows the Eq. 9 and then we let $\theta_c \in \mathbb{R}^k$ be the k-dimensional word vector for encoding.

$$\begin{aligned} S(C_j) &= \sum_{w \in U} p(C_j|w) \\ \theta_c &= \max_{C_j}(S(C_j)), C_j \in C \end{aligned} \tag{9}$$

Here, C is the set of concepts.

Concept Sensitive Model Fig. 2 shows the framework of Concept Sensitive NHG model with a multi-layer Bi-LSTM encoder. The function $f(x)$ represents the keyword extraction from the input text and the function $g(k)$ represents the acquisition of concept information. In our model, the concept information of the input text is entered into the encoder along with source text, which will affect weight matrices in encoder. After encoder step, we obtain the output of hidden states $\mathbf{h}$ which contains the concept information. And we used $\mathbf{h}$ with attention mechanism in decoder to generate headline.

4 Experiments

The hypothesis of our model are: (1) The multi-layer encoder is more effective for information compression and language generation. (2) Concept information helps to generate better headlines and improve the results of multi-layer Bi-LSTM encoder model. We experimented the proposed CNHG model on the task of headline generation to verify our hypotheses.

4.1 Datasets and Evaluation Metrics

To demonstrate the effectiveness of our method, we used the English Gigaword Fifth Edition [7] corpus, which contains 9.5 million news articles with corresponding headlines[2] from various news services. We processed the data in the same way as [5], as a consequence, about 4 million examples are selected as our training set[3].

We evaluated our models on the DUC-2004 dataset[4]. It consists of 500 article-headline pairs, and each paired with 4 human-generated reference headlines. In addition, we also used Gigaword test data[5] used in [5] as our test set.

In this series of experiments, we utilized several versions of ROUGE [9] to evaluate the performance of headline generation. Similar to [5, 8, 18], for DUC-2004 test set, we reported 75 bytes capped (use the limited length with 75 bytes) recall scores of ROUGE-1, ROUGE-2 and ROUGE-L to evaluate our systems. And similar to [8, 18], for Gigaword test set, we used full-length F1 scores of ROUGE-1, ROUGE-2 and ROUGE-L to evaluate our systems. In previous work, limited length recall was widely used, but that make it difficult for researchers to compare their results because of the choice of length limit varies with the corpus. The average length of headline in Gigaword test set is 8.3 words, that is significantly shorter than the summary in DUC-2004, and a shorter summary tends to get lower recall score. On the contrary, Full-length F1 makes evaluation unbiased to summary length and can penalize a longer summary. Thus, when testing on Gigaword test set, we report the full-length F1 scores.

[2] We paired the first sentence of each article with its headline to form sentence-headline pairs. And Then we used the PTB tokenization to preprocess the pairs with tokenziation.

[3] The splits of Gigaword for training can be found at https://github.com/facebook/NAMAS

[4] It can be downloaded from http://duc.nist.gov/ with permission

[5] It can be obtained from https://github.com/harvardnlp/sent-summary

4.2 Implementation Details

For all the models we discuss below, the word embeddings are randomly initialized with 128 dimension and then updated during training. We used stochastic gradient descent with mini-batches of 64 to minimize our loss and randomly shuffled the training data at every epoch. Besides, we set the vocabulary size to $150,000$ and the hidden unit size to 256. Since the vocabulary size is too large, we used sampled softmax method with the value of 4096 to speed up the training. We did not utilize any dropout or regularization, but gradient clipping used. For all our models, we initialized the learning rate to 0.15, and decayed the learning rate every $30,000$ batches with the decay rate to 0.95.

We trained all our models on a single GeForce GTX TITAN GPU. For 4-NHG it takes about 20 hours for an epoch. For CNHG it takes about 24 hours. At decoder time, we used beam search of size 8 to generate the headline, using a batch size of 1.

5 Results and Analyses

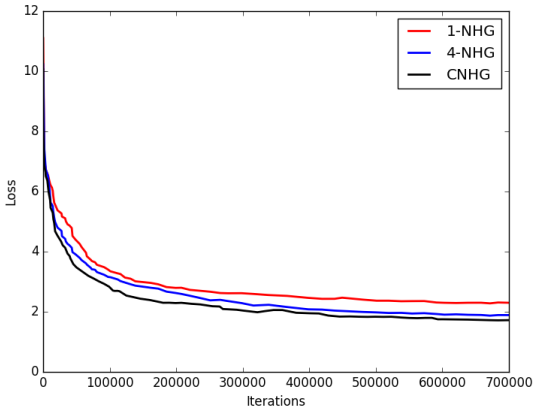

Fig. 3. Loss vs iteration

Fig. 3 shows the training loss of iterations, 1-NHG represents a mono-layer Bi-LSTM encoder model. At the beginning of training, the convergence speed of three models is similar, but gradually the CNHG model converges faster than the 4-NHG model and 1-NHG. When the number of iterations reaches about $200K$, the convergence speed of three models becomes slow and eventually tends to be stable. From the final training loss values, it can be seen that the CNHG model converges best and 4-NHG obtains better training effects than 1-NHG.

Table 1 shows the results of 1-NHG and 4-NHG models on Gigaword English test set. AS we can see 4-NHG model outperforms 1-NHG model on Gigaword test set and achieves significant improvements. Combined with Fig. 3, we found that the multi-layer encoder is more effective for information compression and language generation than mono-layer encoder, which supports the hypothesis 1.

The evaluation results on Gigaword and DUC-2004 test sets are presented in Table 2. The baseline MOSES+ generates headline based on MOSES [16], which is a

Table 1. Results of 1-NHG and 4-NHG models on Gigaword English test set.

	Gigaword		
model	ROUGE-1	ROUGE-2	ROUGE-L
1-NHG	33.59	13.84	31.13
4-NHG	**35.03**	**14.97**	**32.87**

phrase-based machine translation system. ABS and ABS+ are an attention-based neural headline generation system of [5], and ABS+ is an enhanced version of ABS. First note that the baseline ABS+ performs better than ABS and MOSES+ on both test sets except for the F1-score of ROUGE-2 in Gigaword. Both RAS-Elman and RAS-LSTM [8] which utilize a convolutional encoder and an attention-based decoder achieve statistically significant improvements from ABS+ for all three variants of ROUGE on Gigaword. The state-of-the-art baseline BWL [18] , namely words-lvt5k-lsent, outperforms other baselines on both test sets.

Table 2. Results of our 4-NHG and CNHG models against other baselines on DUC-2004 and Gigaword English test sets.

	DUC-2004			Gigaword		
model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
MOSES+	26.50	8.13	22.85	28.77	12.10	26.44
ABS	26.55	7.06	22.05	29.55	11.32	26.42
ABS+	28.18	8.49	23.81	29.76	11.88	26.96
RAS-Elman	**28.97**	8.26	24.06	33.78	15.97	31.15
RAS-LSTM	27.41	7.69	23.06	32.55	14.70	30.03
BWL	28.61	9.42	25.24	35.30	**16.64**	32.62
4-NHG	27.31	8.68	24.74	35.03	14.97	32.87
CNHG	28.10	**9.56**	**25.71**	**35.51**	15.54	**33.38**

The results of our models is exciting and inspiring. For Gigaword corpus, we reported the F1-score of ROUGE-1, ROUGE-2, and ROUGE-L. AS we can see, our CNHG model outperforms 4-NHG model on all variants of ROUGE and improves the ROUGE scores up 0.5 points compared with 4-NHG model. Besides, our CNHG model outperforms the start-of-the-art baseline BWL on two of three variants of ROUGE on Gigaword test set, while being competitive on ROUGE-2. For DUC-2004 corpus, we computed the recall score of ROUGE-1, ROUGE-2, and ROUGE-L. The results show that our CNHG model improves the ROUGE scores up 1 points compared with 4-NHG model and outperforms BWL on ROUGE-2, ROUGE-L. In general, our CNHG model significantly and consistently improves the performance of 4-NHG, and outperforms the state-of-art system on both test sets, which also support our hypothesis 2.

Fig. 4 shows a soft alignment between the input text and the generated headline. From this we see which words in input text were considered more important when generating the target word. For example, the word “fms” in headline depends on the phrase “foreign ministers” in input text.

Table 3. Examples of original articles, reference headlines and generated headlines by our models on Gigaword test set.

Article(1):	arbitrary arrests , torture , prisoners dying in detention and the death penalty are current practices in guinea , human rights organization amnesty international said thursday in a report published here .
Reference:	amnesty deplores human rights violations in guinea
4-NHG:	amnesty international condemns torture in guinea
CNHG:	amnesty slams human rights abuses in guinea
Article(2):	burma has put five cities on a security alert after religious unrest involving buddhists and moslems in the northern city of mandalay , an informed source said wednesday .
Reference:	burma puts five cities on security alert after religious unrest
4-NHG:	burma puts five cities on security alert
CNHG:	burma puts five cities on alert after religious unrest
Article(3):	secretary of state warren christopher widened consultations on an israeli-lebanese ceasefire wednesday by including egypt and saudi arabia in the effort , an official said .
Reference:	christopher widens consultations over israel-lebanon crisis
4-NHG:	christopher widens mideast talks with saudi arabia
CNHG:	christopher widens consultations on israel-lebanon ceasefire
Article(4):	the leader of germany 's jews , ignatz bubis , urged the government on friday to at least symbolically back an industry initiative to establish a fund for nazi slave laborers .
Reference:	jewish leader calls on government to establish fund for nazi slave
4-NHG:	jewish leader urges government to fund nazi slave fund
CNHG:	jewish leader urges government to open fund for slave laborers
Article(5):	up to ## afghans have been killed and hundreds injured by a massive explosion at an ammunition depot in the eastern provincial capital jalalabad , kabul red cross officials said thursday .
Reference:	up to ## killed in afghan blast accident suspected by terence white
4-NHG:	explosion at ammunition depot kills up to ## afghans
CNHG:	at least ## dead and hundreds injured in afghan blast
Article(6):	the eighth asia-pacific traditional arts festival opened saturday at the center for traditional arts in eastern yilan county saturday , featuring folk music , dance and theater groups from the mekong river region of indochina .
Reference:	traditional arts center in yilan presents mekong art
4-NHG:	#th asia-pacific arts festival opens in eastern china
CNHG:	#th asia-pacific traditional arts festival opens

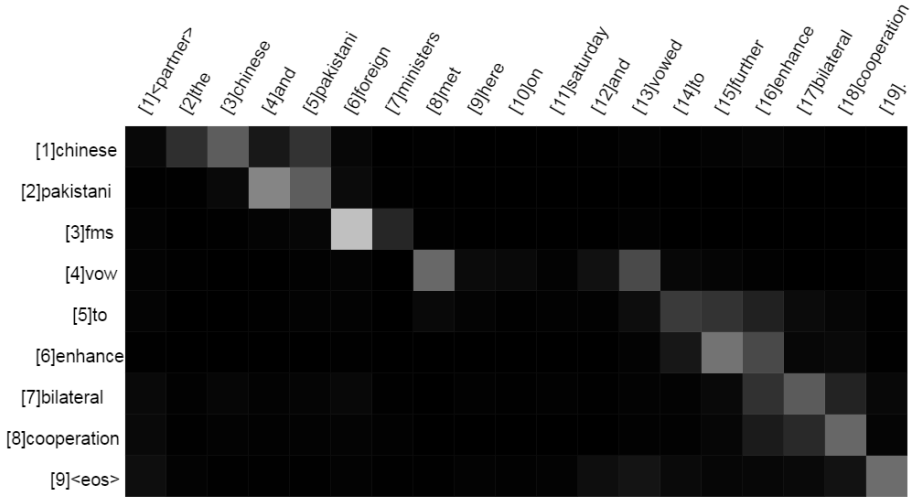

Fig. 4. A sample alignment generated by CNHG. The x-axis and y-axis of each plot correspond to the words in the input and the generated headline, respectively. The rows represent the distribution over the input for each generated word.

Finally, in Table 3 we present several anecdotal examples of headlines by our models on Gigaword test set for discussion. We can observe that our CNHG model is capable of capturing the semantically important words of the input. For instance in Article 1, CNHG can successful use "human rights" to summarize "arbitrary arrests, torture, prisoners dying" while 4-NHG only use "torture", and in Article 2 it generates headline with important information "religious unrest". Compared to the true headline, CNHG can use some of words that are different from reference to form a headline without changing the original meaning. For Article 3, CNHG uses word "ceasefire" while reference uses "crisis", and in Article 4 it uses word "open" not "establish". Despite our models capture a coherent and meaningful headline but differs from the true headline, however, the score of ROUGH is low. As shown in Article 5, two models capture the main information to generate headlines but the score of ROUGE is not high. On the other hand, our models sometimes pay attention to the wrong area in the article and generate an inferior headline as shown in Article 6.

6 Conclusion

In this paper, we proposed a concept sensitive NHG model for headline generation. In the model, we used a multi-layer Bi-LSTM encoder with attention mechanism to automatically generate coherent headlines. To enable the generated headlines fulfill the core concept from a document, the proposed model fed the extracted concepts into the encoder, which can be treated as a guidance to generate focused and salient headlines. The experimental results of headline generation on both Gigaword and DUC-2004 dataset show that our CNHG model outperforms the start-of-the-art models.

Acknowledgments. The work was supported by National Basic Research Program of China (973 Program, Grant No.2013CB329303), National Nature Science Foundation of China (Grant No.61602036), Beijing Advanced Innovation Center for Imaging Technology (BAICIT-2016007).

References

1. Dorr., Bonnie., David Zajic., Richard Schwartz.: Hedge trimmer: A parse-and-trim approach to headline generation. Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5. Association for Computational Linguistics (2003)
2. Sutskever, I., Vinyals, O., Le, Q. V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems (2014)
3. Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Computer Science (2014)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Computer Science (2014)
5. Rush, A. M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. Computer Science (2015)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation, 9(8), 1735 (1997)
7. Parker, R., Graff, D., Kong, J., Chen, K., Maeda, K.: English gigaword fifth edition. (2011)
8. Chopra, S., Auli, M., Rush, A. M.: Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.93-98 (2016)
9. Flick, C.: ROUGE: A Package for Automatic Evaluation of summaries. The Workshop on Text Summarization Branches Out, pp.10 (2004)
10. Over, P., Dang, H., Harman, D.: Duc in context. Information Processing and Management, 43(6), 1506-1520 (2007)
11. Zajic, D., Dorr, B., Schwartz, R.: BBN/UMD at DUC-2004: Topiary. Document Understanding Conference at Nlt/naacl, pp.112–119 (2004)
12. Cohn, T., Lapata, M.: Sentence compression beyond word deletion. In: COLING 2008, International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, Uk Vol.163, pp.137-144 (2008)
13. Woodsend, K., Feng, Y., Lapata, M.: Title Generation with Quasi-Synchronous Grammar. In: Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, Mit Stata Center, Massachusetts, Usa, A Meeting of Sigdat, A Special Interest Group of the ACL, pp.513-523 (2010)
14. Takase, S., Suzuki, J., Okazaki, N., Hirao, T., Nagata, M.: Neural Headline Generation on Abstract Meaning Representation. In: Conference on Empirical Methods in Natural Language Processing, pp.1054-1059 (2016)
15. Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., Bengio, Y.: Pointing the Unknown Words. In: Meeting of the Association for Computational Linguistics, pp.140-149 (2016)
16. Koehn, Philipp, Hoang, Hieu, Alexandra, CallisonBurch, et al.: Moses: open source toolkit for statistical machine translation. In: in Proceedings of the Association for Computational Linguistics ACL 07, 9(1), 177–180 (2007)
17. Ayana, Shen, S., Liu, Z., Sun, M.: Neural headline generation with minimum risk training. (2016)
18. Nallapati, R., Zhou, B., Santos, C. N. D., Gulcehre, C., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. (2016)