A Hierarchical Model for Text Autosummarization

Zhenpeng Zhou
Stanford University
Stanford, CA 94305
zhenpeng@stanford.edu

Abstract

Summarization is an important challange in natural language processing. Deep learning methods, however, have not been widely used in text summarization, although neural networks have been proved to be powerful in natural language processing. In this paper, an encoder-decoder neural network model is applied to text summarization, as an important step toward this task. Besides, a hierarchical model, which builds the sentence representations and then paragraph representations, enables the summarization for long documents.

1 Introduction

Summarization is an important challenge of natural language processing. [1, 2]Most summarization systems are *extractive* methods.[3] However, extractive methods are limited by their nature, as summaries are not essentially come from the source. Comparatively, *abstractive* methods are supposed to generated summaries from the source files, although they may not appear as part of the original.[4]

Recently, deep learning methods have been proven to be promising in gereating representations and language models. [4, 5, 6, 7, 8] Convolutional neural network (CNN), and recurrent neural network (RNN) are powerful in learning representations of texts. However, the understanding of long documents is still far from satisfactory.[9]

In this paper, An encoder-decoder model is used to summarize a news article into its title. More specificly, a hierarchical LSTM [10] is used as an encoder, in which the representations of sentences are learned by a LSTM model whose inputs are words, and the representation of the document is learned by another LSTM model whose inputs are sentences representations. A normal LSTM is used as a decoder.

2 Related Work

A lot of works in summarization are extractive methods, which are using word frequency to determined the importance of the word, in order to select senteces. [11, 12, 13, 14]

Abstractive methods, however, are more similar to the way of human beings generating summaries. Abstractive sentence summarization has been traditionally connected to the task of headline generation. Banko et al. [15] showed work using statistical machine translation directly for abstractive summarization. Cohn and Lapata [16] give a tree transduction compression method with a max-margin learning algorithm.

Deep learning methods provided a framework for data-driven approach of generating summaries. In [17], a recurrent neural network with attention mechanism was built to generate summaries. Rush et al. [7] use convolutional models with attention mechanism, showing state-of-art performance on DUC tasks. Hu et al. [18] proposed a large data set for Chinese summarization, with recurrent neural network as encoder and decoder.

3 Model Architecture

3.1 Long-Short Term Memory (LSTM)

Long Short Term Memory networks, usually just called "LSTMs", are a special kind of RNN, which is capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber [19]. They work tremendously well on a large variety of problems, and are now widely used.

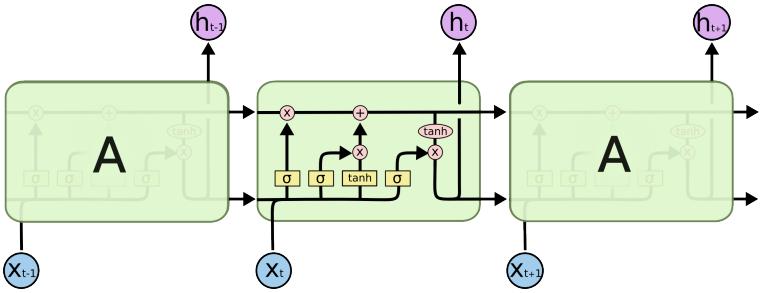


Figure 1: The illustration of a LSTM cell, from `http://colah.github.io/posts/2015-08-Understanding-LSTMs`

The LSTM cell is illustrated in Figure 1, which can be simplified as

$$r = \text{LSTM}(\{x_t\})$$

where x_t is the input of LSTM cell at time step t, r is the output.

3.2 Normal LSTM

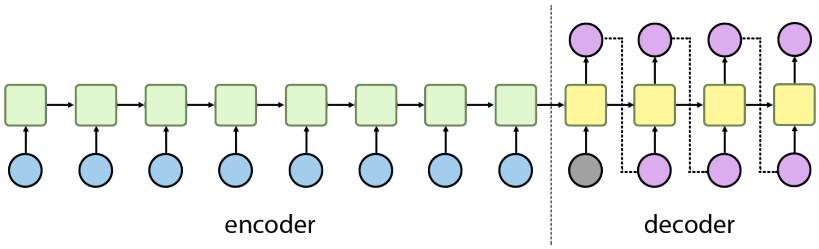


Figure 2: The illustration of the normal LSTM encoder-decoder model.

A normal LSTM as encoder and decoder is used as a baseline model. The encoder consists of a bi-directional LSTM, while the decoder consists of a uni-directional LSTM. As shown in Figure 2.

3.3 Hierarchical LSTM

A hierarchical LSTM [10] is used as an encoder. A LSTM model is used as at sentence level. For a given sentence of $\text{sent}_i = \{w_i | i = 1 \cdots n\}$, in which w_i is the ith word in the sentence. the LSTM cell is applied on the sentence recurrently, the sentence representation is taken to be the last hidden state h_n of the LSTM output.

$$r_{\text{sent}} = \text{LSTM}(\{w_i\})$$

Another LSTM model is used at the document level. The representations of sentences are then feeded into the LSTM cell, the last hidden state of h_m is take to be the document representation.

$$r_{\text{doc}} = \text{LSTM}(\{sent_i\})$$

LSTM decoder is used to generate word sequence. The LSTM output at time $t-1$ is used as the input at time t.

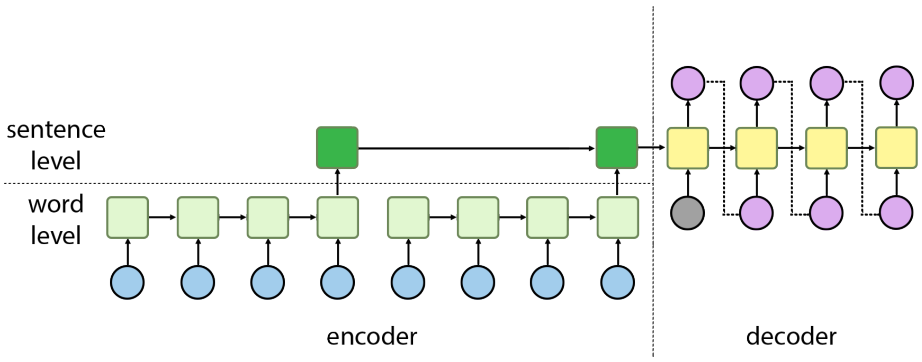


Figure 3: The illustration of the hierarchical LSTM encoder-decoder model.

The hierarchical LSTM model is shown in Figure 3.

4 Experiments

4.1 Dataset

The Signal Media One-Million News Articles Dataset is used for training and testing. The first three paragraphs is used as original document, and the title is used as summarization target. Minimal preprocessing step is applied including lower-casing, and replacing tokens less than 10 times with an <unknown> label. Samples with more than 20% <unknown>s are dropped out, giving a final dataset of 700K samples. The mean length of the inputs is 154 words, with a standard deviation of 51 words.

Dataset is seperated in to training set, cross validation set and test set, with a ratio of 0.70, 0.15, and 0.15.

4.2 Training

The word embedding matrix is initialized with word vectors trained on the dataset by GloVe. [20] Bucketing, which groups inputs of the same length together, is used when unrolling the LSTM, in order to enable minimal padding and speed up training. Adam [21] is used as optimizer for training.

During decoding, Beam search of size 5 is used to generate the summary.

4.3 Evaluation

the F1-score of ROUGE-1, and ROUGE-2 [22] on the test set are used as evaluations of results.

4.4 Experiments of models

Three models of

- `l2_norm_lstm`: Two-layer normal LSTM model.
- `l2_hier_lstm`: Two-layer hierarchical LSTM model, the encoders at both sentence level and document level consist of 2 layers.
- `l4_norm_lstm`: Four layer normal LSTM model.

are trained on the same dataset with the same hyperparameter.

5 Results

5.1 ROUGE score on Signal Media Dataset

F-1 scores of ROUGE-1, and ROUGE-2 are used as metrics of evaluation, The ROUGE score is defined as

$$\text{ROUGE} - \text{N} = \frac{\sum_{S \in \{\text{RefSummaries}\}} \sum_{gram_n \in S} \text{count}_{match}\left(gram_n\right)}{\sum_{S \in \{\text{RefSummaries}\}} \sum_{gram_n \in S} \text{count}\left(gram_n\right)}$$

Table 1: The performance of the methods on Signal Media Dataset

Model	ROUGE-1	ROUGE-2
l2_norm_lstm	28.67	9.58
l2_hier_lstm	**32.62**	16.13
l4_norm_lstm	32.27	**16.39**

The ROUGE scores of the test set is shown in Table 1, which show that the two-layer hierarchical LSTM model is significantly better than two-layer normal LSTM, but has a similar performance to the four-layer normal LSTM. This is supposed to attributed to the fact that a hierarchical LSTM model can be regarded as a double-layered normal LSTM model with some connections disabled, as shown in Figure 4. This disabled connections make training process faster. The training speed of two-layer hierarchical LSTM is 30% faster than that of four-layer normal LSTM.

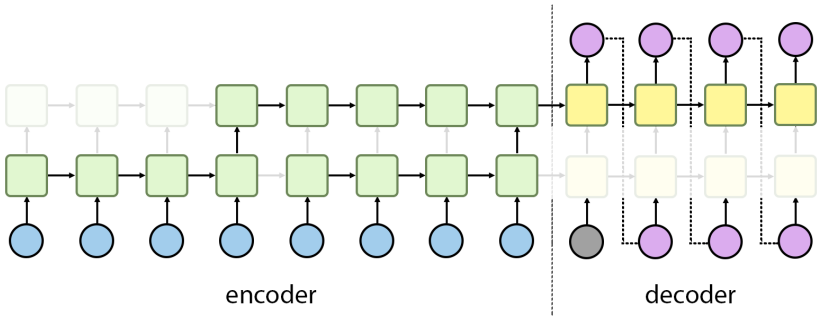


Figure 4: The similarity between a hierarchical LSTM model and a double-layered normal LSTM model, The normal lstm can be converted to hierarchical LSTM with the connections in light colors disabled.

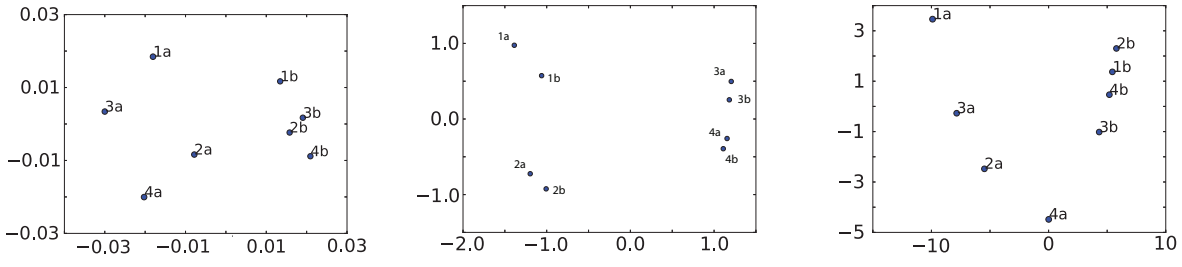
5.2 ROUGE Score on DUC-2004 Dataset

Table 2: The performance of the methods on DUC-2004 Dataset

Model	ROUGE-1	ROUGE-2
ABS+	**28.18**	**8.49**
l2_norm_lstm	25.07	6.14
l2_hier_lstm	27.63	7.68
l4_norm_lstm	27.80	7.45

In Tabel 2, The performance of our model on the DUC-2004 dataset is also compared with the ABS+ [7], which is the state-of-art model on that dataset. A similar performace was achieved with 2-layer hierarchical LSTM and 4-layer normal LSTM models.

5.3 Vector representations of paragraphs



l2_norm_lstm l2_hier_lstm l4_norm_lstm

The sample short paragraphs:

para_1a: I like the car. She likes the car.	para_1b: She likes the car. I like the car.
para_2a: I like the car. She doesn't like the car.	para_2b: She doesn't like the car. I like the car.
para_3a: I don't like the car. She likes the car.	para_3b: She likes the car. I don't like the car.
para_4a: I don't like the car. She doesn't like the car.	para_4b: She doesn't like the car. I don't like the car.

Figure 5: The PCA projection of the three model outputs of sample short paragraphs.

The neural network encoder is capable of learning the vector representations of paragraphs. Figure 5 shows the PCA projection of the three models' encoder outputs of some sample paragraphs. In the samples, para_ia and para_ib are paragraphs with same meanings but different sentence orders, while para_ia and para_ja are paragraphs with different meanings. For the 2-layer hierarchical LSTM model, changing the orders of sentences does little changes to the meanings, while changing the words varies the meanings a lot. Both the 2-layer and 4-layer normal LSTM can differentiate the sample short paragraphs, but cannot attribute the paragraphs with same sentences but different order to similar meanings. On the other hand, for 2-layer normal LSTM model, the distances between the paragraph vectors are comparatively short, indicating that shallow LSTMs are not powerful enough to learn the meanings of paragraphs.

5.4 Sample Summarizations

The Sample summaries by the hierarchical model on the test set are shown in Table 3, from which we can see the model can generate meaningful and relavent summaries. On the other hand, this model performs badly when the documents have complicated sentences or many less frequent words.

6 Conclusion

In this work, a hierarchical LSTM encoder-decoder model is proposed for abstractive (long) documents summarization, giving promising results. This model is also capable of generating paragraph and document representations. Besides, the performance and relationship between hierarchical LSTM and multi-layer LSTM are also discussed.

In the future, attention mechanism could be added to the model to achieve better performance, and more LSTM layers could be applied to understand the sentence better.

In terms of natural language processing, most problems can be transformed to the problem of getting representations of language elements (words, sentences, paragraphs, and documents). The work of word to vector was well developed. However, the sentence to vector is still far from satisfactory. RNN models on natural language processing is still not powerful enough like CNN models on computer vision. This might because the LSTM models are not good enough to model sentences, or the sentences can be modeled by LSTM, but the neural network is not "deep" enough to understand

Good summaries
D: listening to politicians campaigning what you get is that after october , <unk> will be living in heaven . candidates empty promises paints a flawless tanzania in which no one will labour for anything . if you critically look at things which politicians in the campaign trail promise they are going to accomplish if elected , you will be petrified . **O**: my take on this : politicians should give us a break on empty promises. **G**: how politicians are giving <unk> empty promises
D: official says number ' of emails copyright 2015 cable news network/turner broadcasting system , inc. all rights reserved . this material may not be published , broadcast , rewritten , or redistributed . an email chain between former secretary of state hillary clinton and of u.s. central command david petraeus from january and february 2009 is raising questions about whether some of the emails on clinton 's private email server are mistakenly deemed personal and not included among the 55,000 pages of emails she turned over to the state department . **O**: new hillary clinton email chain discovered **G**: hillary clinton email service discovered
Bad Summaries
D: new product gives marketers access to real keywords , conversions and results along with 13 months of historical data san francisco , ca – (marketwired) – 09/17/15 – <unk> , a marketing analytics company that uses distinctive data sources to paint a complete picture of the online customer journey , today announced the launch of <unk> elite , giving marketers insight into what their customers are doing the 99 % of the time they 're not on your site . for years , marketers have been unable to see what organic and paid search terms users are entering , much less tie those searches to purchases . <unk> not only injects that user search visibility back into the market , but also makes it possible to tie those keywords to conversions – for any web site . **O**: <unk> gives marketers renewed visibility into paid and organic keywords with launch of <unk> elite **G**: watch firm you renewed visibility with modern and more keywords
D: nhs patients to be given option of travelling to calais for surgical procedures nhs patients in kent are set to be offered the choice of travelling to calais for surgical treatments , local health commissioners have confirmed . **O**: nhs patients to be given option of travelling to calais for surgical procedures **G**: outside owner be given of travelling to school after surgical procedures

Table 3: Examples of generated summaries from the hierarchical LSTM model on the test set. **D**: source document, **O**: original title, **G**: generated summaries.

them. The former problem might be solved by adding hidden variables to neural networks, like combining RNN and hidden Markov models. The latter problem might be solved by combing residue learing with LSTM, to achieve a deep RNN for better understanding of sentences.

References

[1] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[2] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[3] Misha Denil, Alban Demiraj, and Nando de Freitas. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*, 2014.

[4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[5] Pengfei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuanjing Huang. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *Proceedings*

of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2326–2335. Citeseer, 2015.

[6] G PadmaPriya and K Duraiswamy. An approach for text summarization using deep learning algorithm. *J Comput Sci*, 10:1–9, 2014.

[7] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[9] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[10] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[11] Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence*, pages 205–215. Springer, 2002.

[12] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[13] Kam-Fai Wong, Mingli Wu, and Wenjie Li. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics, 2008.

[14] Katja Filippova and Yasemin Altun. Overcoming the lack of parallel data in sentence compression. In *EMNLP*, pages 1481–1491. Citeseer, 2013.

[15] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325. Association for Computational Linguistics, 2000.

[16] Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics, 2008.

[17] Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. Abstractive sentence summarization with attentive recurrent neural networks.

[18] Baotian Hu, Qingcai Chen, and Fangze Zhu. Lcsts: a large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865*, 2015.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.