



# Order-Preserving Abstractive Summarization for Spoken Content Based on Connectionist Temporal Classification

Bo-Ru Lu, Frank Shyu, Yun-Nung Chen, Hung-Yi Lee, and Lin-Shan Lee

College of Electrical Engineering and Computer Science, National Taiwan University, Taiwan  
{r05942070, b01901051, yvchen, hungyilee}@ntu.edu.tw, lslee@gate.sinica.edu.tw

## Abstract

Connectionist temporal classification (CTC) is a powerful approach for sequence-to-sequence learning, and has been popularly used in speech recognition. The central ideas of CTC include adding a label “blank” during training. With this mechanism, CTC eliminates the need of segment alignment, and hence has been applied to various sequence-to-sequence learning problems. In this work, we applied CTC to abstractive summarization for spoken content. The “blank” in this case implies the corresponding input data are less important or noisy; thus it can be ignored. This approach was shown to outperform the existing methods in term of ROUGE scores over Chinese Gigaword and MATBN corpora. This approach also has the nice property that the ordering of words or characters in the input documents can be better preserved in the generated summaries. **Index Terms:** abstractive summarization, headline generation, connectionist temporal classification (CTC)

## 1. Introduction

Spoken content summarization is to generate a summary that describes the core ideas of the spoken document [1], which has been performed on several types of spoken content, including news [2, 3, 4], lectures [5, 6, 7] and meeting records [8, 9, 10]. “Abstractive” summarization, usually referred to as the “headline generation”, means that the headline is not necessarily directly extracted from the input spoken document but automatically generated. When conducting headline generation [11, 12, 13, 14], the model must be capable of both extracting and describing the key concepts of the input. We can formulate this task as a sequence-to-sequence problem, with the machine-transcribed spoken document being the input sequence in the first-stage development, and the summary or headline being the output sentence. This task is important for spoken content, because spoken content is difficult to be shown on the screen, let alone scanned and comprehended by user. Summarization is therefore very helpful for browsing the spoken content.

Various approaches have been proposed for sequence-to-sequence problems, such as speech recognition [15, 16], neural machine translation [17, 18], and dialogue modeling [19, 20]. Among various deep-learning-based methods, Cho *et al.* [21] first introduced a basic sequence-to-sequence model based on recurrent neural networks (RNNs) encoder-decoder architecture [22, 23, 24, 25], namely Seq2Seq. An attention mechanism was introduced by Bahdabau *et al.* [26], which allows the decoder to directly fetch the information from input. In Seq2Seq models, an input sequence is encoded into a fixed-length “sentence vector” and then decoded into an output sequence. However, this method loses track of the ordering of the elements in the input sequence, since the entire input sequence is encoded into a single “sentence vector”. In other words, the ordering of the input elements does not directly affect that of the output la-

bels. Order-preserving is desired here, because very often the word order in a spoken document has to do with its semantics.

This paper proposes a connectionist temporal classification (CTC) [15, 27, 28, 29] based model for spoken content summarization to address the above order-preserving issue. We apply the CTC loss function to impose a hard constraint to preserve the order when generating the output sequence. Unlike the Seq2Seq method that decodes the output sequence after the entire input sequence is read, the CTC loss function decodes the output sequence character by character. That is, the words appearing at the beginning of the input sequence should appear at the beginning of the output sequence if found important. This property of CTC loss function makes it attractive for spoken content summarization. This approach was evaluated on the Chinese Gigaword [30] and the MATBN [31] corpora of Mandarin broadcast news using ROUGE scores metrics [32]. The results showed that the proposed model significantly outperformed the state-of-the-art approaches [33].

## 2. Model

The proposed model consists of an RNN and a CTC model, where the input is machine-transcribed spoken documents and the output is their summaries. The input sequence is segmented into characters or words when feeding into the RNN. Let  $x_t$  denote the input element at time  $t$  and the input sequence is then  $X = [x_1, x_2, \dots, x_T]$ , where  $T$  is the input sequence length. The RNN outputs a probability distribution vector  $y_t$  for each time step  $t$ , so the output sequence is  $Y = [y_1, y_2, \dots, y_T]$ . Finally, the output headline, denoted by  $Z = [z_1, z_2, \dots, z_U]$ , is generated by a forward-backward algorithm that calculates the optimal path based on  $Y$ . Note that the sequence length of  $Z$ , denoted by  $U$  here, should be significantly less than  $T$ , because  $Z$  is a summary or a headline. Below, we briefly introduce CTC and how it is applied to the summarization task as shown in Figure 1.

### 2.1. Connectionist Temporal Classification (CTC)

The connectionist temporal classification (CTC) training is based on a loss function in training RNN, which maps an input sequence  $X = [x_1, x_2, \dots, x_T]$  onto a target sequence  $Z = [z_1, z_2, \dots, z_U]$ . The CTC is designed for sequence learning problems that have input and output sequences with different lengths, namely  $U \leq T$ . For instance, in speech recognition, an acoustic sequence is transcribed into a phoneme sequence. The outputted phoneme sequence is obviously shorter than the acoustic frame sequence.

In CTC training, an extra label “blank” (denoted by “-”) is inserted between every two adjacent output labels  $z_u$  and  $z_{u+1}$  in the target sequence  $Z^*$ . Hence, the output elements have  $L' = L + 1$  labels, where  $L$  is the number of distinct out-

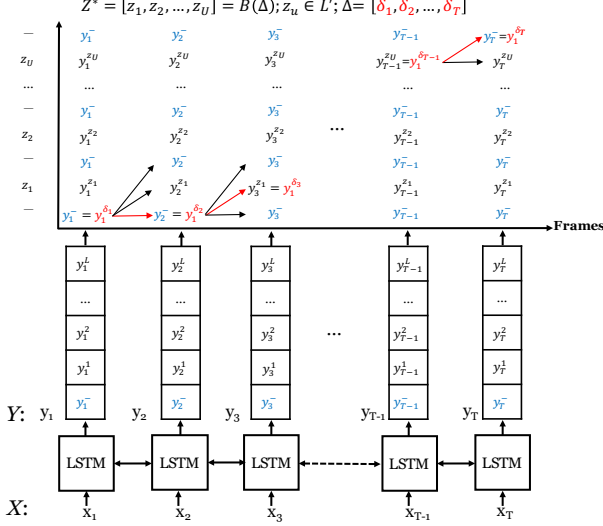


Figure 1: Illustration of the CTC training. For a given input sequence  $X = [x_1, x_2, \dots, x_T]$  at the bottom, a sequence of vectors  $Y = [y_1, y_2, \dots, y_T]$  is generated, where  $y_t$  is a probability distribution over all possible output labels plus the blank “-”, as shown in the lower part of the figure. The upper part shows a map for finding the probability of a given target sequence  $Z^* = [z_1, z_2, \dots, z_U]$  for all possible output sequence hypotheses  $\Delta = [\delta_1, \delta_2, \dots, \delta_T]$  based on  $Y$ . We use the output sequence hypothesis  $\Delta$  to interpret here, highlighted in red. All blanks “-” here are highlighted in blue.

put labels. Each element in  $Y$ ,  $y_t$ , is an  $L'$ -dimension vector,  $y_t = [y_t^n, n = 1, 2, \dots, L']$ , where  $y_t^n$  is the probability of observing the  $n$ -th output label at time index  $t$ . This is shown in the lower part of Figure 1. Given  $Y$ , we can then calculate the probability for an arbitrary output label sequence, denoted by  $\Delta$ . An output label sequence is regarded as a sequence hypothesis used to generate the final output sequence  $Z^*$ . This sequence hypothesis is represented by  $\Delta = [\delta_1, \delta_2, \dots, \delta_T]$ , where  $\delta_t$  is an output label at time index  $t$ .  $\Delta$  is a sequence whose length equal to that of  $Y$ , but can be mapped onto the output sequence  $Z^*$  with the shorter length by a function  $\mathcal{B}(\cdot)$ . The function  $\mathcal{B}(\cdot)$  is a many-to-one mapping function, which removes the repeated labels and “-” from  $\Delta$ , but if two identical labels are separated by a “-”,  $\mathcal{B}(\cdot)$  generates 2 consecutive labels in the output. For instance, if  $\Delta = [\alpha, \alpha, \text{“-”}, \text{“-”}, \beta, \text{“-”}, \beta]$ , then  $Z^* = \mathcal{B}(\Delta) = [\alpha, \beta, \beta]$ .

The probability of each output sequence hypothesis  $\Delta$  can be expressed by

$$P(\Delta | X) = \prod_{t=1}^T y_t^{\delta_t}, \quad (1)$$

where  $y_t^{\delta_t}$  is the probability of observing the output label  $\delta_t$  at time index  $t$  as shown in the upper half of Figure 1. During training, given a ground truth target sequence  $Z^*$ , the probability of  $Z^*$  given  $X$  can be calculated by

$$P(Z^* | X) = \sum_{\Delta \in \mathcal{B}^{-1}(Z^*)} P(\Delta | X). \quad (2)$$

Because  $\mathcal{B}(\cdot)$  is a many-to-one function, many sequence hypotheses  $\Delta$  may lead to the same ground truth sequence  $Z^*$ .

This is why the summation in (2) is over the set of all such sequence hypotheses denoted by  $\mathcal{B}^{-1}(Z^*)$ . The CTC loss function is then defined as in (3),

$$\mathcal{L}_{CTC} = - \sum_{\forall (X, Z^*) \in \theta} \ln P(Z^* | X), \quad (3)$$

where  $(X, Z^*)$  is the pair of the input sequence and its corresponding target sequence, and the  $\theta$  denotes the set of all such pairs in the training set. The probability distribution  $P(Z^* | X)$  can be efficiently solved with the forward-backward algorithm. The network can be trained with gradient descent by taking the derivative of (3).

## 2.2. CTC for Spoken Content Summarization

In CTC, the blank will be removed in the final decoding stage. Therefore, in the summarization task, if an input element  $x_t$  has a high probability that  $\delta_t$  is a blank “-”, the information carried by  $x_t$  may be less relevant to the core concepts of the document.

## 3. Experiments

### 3.1. Experimental Setup

The experiments are performed on the Chinese Gigaword corpus, which consists of 2.2M of headline-document pairs of news stories covering over 284 months from two Chinese newspapers, namely the Xinhua News Agency of China (XIN) and the Central News Agency of Taiwan (CNA). We select the headline-document pairs of the first 278 months as the training set, and the remaining 6 months for testing. This is to test whether the proposed approach is able to predict future headlines given data for the past. This corpus includes only texts but not audio, so the headline-document pairs are good for training. For testing, it is good only for the scenario with zero ASR errors.

In Chinese, a sentence is a word sequence, while a word is composed of one to several characters, and there are no blanks between words in sentences, so a sentence can also be considered as a character sequence. Therefore, both words and characters can be taken as elements of the input/output sequence. The Chinese Gigaword corpus is segmented into sequences of words and characters. We compare the performance between using words and characters as input sequence elements below. Before training the model, both word and character vectors were pretrained with Global Vectors (GloVe) [34]. The numbers and English words are substituted by tags, which give a word-based vocabulary size of 560,772 and a character-based vocabulary size of 6,752.

To evaluate the summarization of spoken content, we further test the CTC network trained with Chinese Gigaword corpus using the MATBN broadcast news corpus in Mandarin Chinese. This corpus consists of 198 hours of broadcast news from the Public Television Service Foundation of Taiwan including corresponding manual transcriptions and human-generated headlines. In addition to the manual transcriptions with 100% accuracy, we also obtain ASR transcriptions using Kaldi toolkit [35] with a character error rate (CER) of 28.72%. The first 126 audio stories are chosen for evaluation. For implementing CTC networks, we use 2 layers of bidirectional LSTM [36, 37] and apply the CTC loss function on the top layer of LSTM. The baselines we compare include the sequence-to-sequence model (Seq2Seq) [21] and the attentive Seq2Seq model [33], where we used 1 layer of bidirectional LSTM as the encoder and 1 layer LSTM as the decoder for both Seq2Seq and Attentive Seq2Seq. Other hyperparameters are tuned for optimization of all models.

Table 1: Performance comparison of various models and input sequence elements over Chinese Gigaword (no ASR errors).

Model		Input	Output	k	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L
Baseline	Seq2Seq	(a) word	character	1	34.47	18.30	8.82	31.26
		(b) character	character	1	36.33	18.58	8.78	32.39
	Attentive Seq2Seq	(c) word	character	1	36.37	20.23	10.23	32.98
		(d) character	character	1	37.97	20.47	10.27	33.88
Proposed: CTC		(e) word	character	1	25.36	9.20	3.43	24.49
		(f) word	character	2	33.58	15.70	7.34	32.20
		(g) character	character	1	<b>42.71</b>	<b>24.62</b>	<b>14.24</b>	<b>40.56</b>

### 3.2. Chinese Sequence Element Preprocessing

As mentioned above, a Chinese word is composed of one to several characters and a Chinese sentence can be considered as a sequence of words or characters. Very often when a Chinese word is composed of several characters, one of them is the “head character” carrying the key information while the others are simply supporting. For example, for the word “美國 (the U.S.)”, the first character “美” is the “head character” and this character alone can also carry the same meaning (the U.S.) as a “mono-character word”. In such cases very often only the “head character” is used in the ground truth headline to make it short. On the other hand, many other Chinese words cannot be reduced to “head characters” because all characters are important. For example, for the word “支持 (support)”, both characters here are important.

With the above background, it is believed that output sequences for summarization better have characters as their elements, because very often for words with “head characters” only the “head characters” are used in the ground truth headlines, and we need to be able to either select the “head characters” or use the whole word when generating the headlines. For the input sequences, it seems that both words and characters can be elements although characters may be preferred since there are much more distinct words than characters. Nevertheless, when words are used as the input sequence elements and characters as the output sequence elements, very often the machine only selects a part of those words without “head characters” to appear in the output sequences. In order to have those words without “head characters”, such as “支持 (support)” mentioned above, appear entirely in the output without being cut and partly deleted, it was found that repeating the same word  $k$  times when entering them to the CTC network was helpful in generating the whole word in the output. This is referred to as “ $k$ -fold input” here. More precisely, for an input sequence consisting of  $N$  words,  $X = [w_1, w_2, \dots, w_N]$ , the  $k$ -fold input for  $k = 2$  is to enter  $X' = [w_1, w_1, w_2, w_2, \dots, w_N, w_N]$  to the CTC network.

### 3.3. Input Element Comparison

We first consider the input sequence elements: words or characters. As mentioned above, when the words are used, the  $k$ -fold input could be applied. The results when the Chinese Gigaword corpus (corresponding to no ASR errors) was used for both training and testing are listed in Table 1. Rows (e)(f) are for words as input sequence elements with  $k = 1$  and 2, while row (g) is for characters as the input sequence elements. We can see  $k = 2$  offered significant improvement (row (f) vs. (e)), while simply using characters is even better (rows (g) vs. (e)(f)). The finding is the same as the statistical claim that most Chinese words are bi-character words [38].

Table 2: Experiment results demonstrating the order-preserving property. The high LCS set accounts for 82.20% of all documents in the test set. The Attentive Seq2Seq is denoted by Att-Seq2Seq. ROUGE-1, ROUGE-2 and ROUGE-L are denoted by R-1, R-2 and R-L, respectively.

Data	Model	R-1	R-2	R-L
High 82.20%	(a) Att-Seq2Seq	41.12	22.66	36.73
	(b) CTC	<b>47.09</b>	<b>27.83</b>	<b>44.91</b>
Low 17.80%	(c) Att-Seq2Seq	<b>23.40</b>	<b>10.29</b>	<b>20.65</b>
	(d) CTC	22.50	9.83	20.50

### 3.4. Comparison with Sequence-to-Sequence Models

We then compare the performance of the proposed CTC network with two strong baselines, sequence-to-sequence model (Seq2Seq) and the attentive Seq2Seq model. The results are listed on the next two sections of Table 1. We see from Table 1, for both Seq2Seq and Attentive Seq2Seq, the cases where characters as the input sequence elements mostly perform better (row (d) vs. (c), (b) vs. (a)). In addition, the attention mechanism clearly helped (rows (c)(d) vs. (a)(b)). The proposed CTC approach with character as input in row (g) achieves the best summarization performance.

### 3.5. Investigation of Order-Preserving Property

CTC networks naturally preserve the ordering of elements from the input sequences and reflect such ordering in the output sequences. Seq2Seq models cannot do so, because they compress the input sequences into fixed length vectors and then decode the vectors. In order to analyze how this property benefits the task, we use the longest common subsequence (LCS) [39] algorithm to evaluate the sequence similarity between the ground truth headline and the input document, giving a real number ranging from 0 to 1. This number is a good indicator of how well the ground truth headline preserves the ordering of the input sequence elements.

We divide the Chinese Gigaword corpus into two groups: one with higher LCS scores and another with lower LCS scores, and then compute their ROUGE scores for two groups. We calculate the LCS scores of all headline-document pairs and find out that 82.20% of the pairs have LCS scores higher than 0.4. Therefore, 0.40 is chosen as a threshold for dividing the groups. In Table 2, the group with low LCS scores has an average LCS score of 28.64%, while the other has an average LCS score of 71.59%. The former group that has high LCS has significantly better performance than the latter group (rows(a)(b) vs. (c)(d)). Within the former group, the proposed CTC network significantly outperforms Attentive Seq2Seq (row (b) vs. (a)), while within the latter group, the proposed CTC network performs

Table 3: An example of generated headlines from various models over an example document in the Chinese Gigaword corpus. The English translation was added here for easy reading.

(a) <b>Text document</b>	美國副國務卿阿米塔吉與日本首相小泉純一郎舉行會談以爭取日本支持美國對伊拉克和北韓的政策這位美國國務院第二號人物禮貌拜會小泉純一郎會談十五分鐘 (The U.S. Deputy Secretary of State Armitage held meetings with Japanese Prime Minister Junichiro Koizumi to fight for Japan’s support about the U.S. policy towards Iraq and North Korea. This second prominent figure of the U.S. State Council visited Junichiro Koizumi for a 15-minute meeting.)
(b) <b>Ground truth</b>	美副國務卿訪日爭取支持對伊拉克政策 (The U.S. Deputy Secretary of State Visited Japan to Fight for Supporting the U.S. Policy toward Iraq)
(c) <b>Seq2Seq</b>	美國務卿與美國務卿舉行會談 (The U.S. Secretary of State Held Meetings with the U.S. Secretary of State)
(d) <b>Attentive Seq2Seq</b>	美國務卿與北韓舉行會談 (The U.S. Secretary of State Held Meetings with North Korea)
(e) <b>Proposed: CTC</b>	美阿米塔日會談支持美對伊拉克北韓政策 (The U.S. Armitage and Japan Held Meetings for Supporting the U.S. Policies toward Iraq and North Korea)

slightly worse than Attentive Seq2Seq (row(d) vs.(c)). Considering that the former group accounts for 82.20% of all test set, the CTC network produces results closer to the ground truth headline when the ground truth headline preserves the ordering better, or the CTC network preserves the ordering better.

### 3.6. Qualitative Analysis

From the generated headlines, we observe that sometimes the proposed model generates the headlines including synonyms that do not appear in the input document, although such a phenomenon is not shown in the example of Table 3. Table 3 is a selected example of the headlines generated by the best model in row (g) of Table 1 for a text document in the Chinese Gigaword corpus. In the part (e), the proposed CTC network compresses the document into a short sentence very close to the ground truth in the part (b) while very well preserving the ordering of characters in the document. The baseline models, Seq2Seq and Attentive Seq2Seq, do not have this capacity as observed in the parts (c)(d). In the part (c), the word “美國務卿 (The U.S. Secretary of State)” appears repeatedly, while in the part (d) the word “北韓 (North Korea)” appears earlier. Although in the parts (c)(d) the keywords are well extracted and grammatically correct sentences are generated, these generated headlines in fact give incorrect semantics that differs from the original document due to the limitation of preserving the ordering for both part (c) and (d). For example, in the part (d) the Attentive Seq2Seq generates a headline “美國務卿與北韓會談 (The U.S. Secretary of State Holds Talks with North Korea)”, which means very differently from the original document because of the reversed word order. In the part (c) the repeated keyword “美國務卿 (The U.S. Secretary of State)” does not make sense. On the contrary, the proposed CTC network in the part (e) is capable of preserving the word ordering and offers the correct semantics with a grammatically correct and short sentence. Also in the part (e) we see the “head character” “美 (The U.S.)” is extracted from the complete word “美國 (The U.S.)” in the original document to be used in the headline, while the complete word “支持 (support)” remains unchanged in the headline because there is no “head character” in this word.

Table 4: Experiment results for the MATBN corpus. The high LCS data accounts for 96.82 % of all documents. The Attentive Seq2Seq is denoted by Att-Seq2Seq. ROUGE-1, ROUGE-2 and ROUGE-L are denoted by R-1, R-2 and R-L, respectively.

Data		Model		R-1	R-2	R-L
Manual	High 96.82%	(a)	Att-Seq2Seq	33.21	15.65	25.02
		(b)	CTC	<b>34.65</b>	<b>19.67</b>	<b>31.82</b>
	Low 3.18%	(c)	Att-Seq2Seq	15.63	<b>7.04</b>	13.41
		(d)	CTC	<b>16.33</b>	5.52	<b>14.03</b>
ASR	High 96.82%	(e)	Att-Seq2Seq	<b>30.74</b>	14.62	23.68
		(f)	CTC	30.55	<b>16.05</b>	<b>27.87</b>
	Low 3.18%	(g)	Att-Seq2Seq	14.94	3.05	11.97
		(h)	CTC	<b>17.42</b>	<b>7.92</b>	<b>17.42</b>

### 3.7. Comparison between Manual and ASR Transcripts

Here we report the results with spoken documents with the MATBN corpus. We select the best proposed model and the best baseline from Table 1, where the selected CTC network is from row (g) of the table and the baseline is Attentive Seq2Seq from row(d). Because this model is actually trained with the first 55 characters of all the documents in Chinese Gigaword, while the news stories in MATBN were much longer, we obtain the headlines for the MATBN news stories in the following way. We use the first 55 characters of the transcriptions to generate the first headline, then shifted the 55-character window by 5 characters (6-th to 60-th characters) to get the next headline. In this way, we generated a total of 20 headlines using the first 150 characters of each news story. We then count the number of characters that appeared in both the generated headline and the first 150 characters in the document for all the 20 headlines. The generated headline maximizing this number is chosen as the final headline.

The results in the upper half of Table 4 are from manual transcriptions, while the lower half for transcriptions obtained with Kaldi including ASR errors. The test set is also divided into two groups according to their LCS scores evaluated with the manual transcriptions. Here the first group of high LCS scores (above 0.42) includes 96.82% of the document-headline pairs. We find that CTC network significantly outperforms Attentive Seq2Seq for documents with high LCS scores (row (b) vs. (a) and (f) vs. (e)), especially in ROUGE-2 and ROUGE-L, for both manual and ASR transcriptions. The ROUGE scores are also much better in the high LCS group (rows (a)(b) vs. (c)(d) and (e)(f) vs. (g)(h)). Moreover, these results clearly show that CTC works equally well with spoken content with ASR errors, and is also good at capturing the ordering of the input sequence on spoken content data sets.

## 4. Conclusions

In this paper, we propose to use a CTC network for abstractive summarization of spoken content. Experiments over Chinese Gigaword and MATBN corpora showed that this approach outperformed the existing sequence-to-sequence learning approaches including that using the attention mechanism. This approach was also shown to be robust with ASR errors, which is a crucial requirement for spoken documents. Overall, the proposed approach is able to preserve the ordering of words/characters in the generated summary, which may be a good reason why it produces better summaries.

## 5. References

- [1] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.
- [2] S. R. Maskey, "Automatic broadcast news speech summarization," Ph.D. dissertation, Columbia University, 2008.
- [3] S. H. Lin, B. Chen, and H. M. Wang, "A comparative study of probabilistic ranking models for chinese spoken document summarization," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 8, no. 1, p. 3, 2009.
- [4] X. Cai and W. Li, "Ranking through clustering: An integrated approach to multi-document summarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1424–1433, 2013.
- [5] Y. N. Chen, Y. Huang, C. F. Yeh, and L. S. Lee, "Spoken lecture summarization by random walk over a graph constructed with automatically extracted key terms," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [6] S. R. Shiang, H. Y. Lee, and L. S. Lee, "Supervised spoken document summarization based on structured support vector machine with utterance clusters as hidden variables." 2013.
- [7] H. Y. Lee, S. R. Shiang, C. F. Yeh, Y. N. Chen, Y. Huang, S. Y. Kong, and L. S. Lee, "Spoken knowledge organization by semantic structuring and a prototype course lecture system for personalized learning," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 5, pp. 883–898, 2014.
- [8] S. Xie, D. Hakkani-Tür, B. Favre, and Y. Liu, "Integrating prosodic features in extractive meeting summarization," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 387–391.
- [9] Y. N. Chen and F. Metze, "Two-layer mutually reinforced random walk for improved multi-party meeting summarization," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 461–466.
- [10] F. Liu and Y. Liu, "Towards abstractive speech summarization: Exploring unsupervised and supervised approaches for spoken utterance compression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1469–1480, 2013.
- [11] M. Banko, V. O. Mittal, and M. J. Witbrock, "Headline generation based on statistical translation," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000, pp. 318–325.
- [12] B. Dorr, D. Zajic, and R. Schwartz, "Hedge trimmer: A parse-and-trim approach to headline generation," in *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*. Association for Computational Linguistics, 2003, pp. 1–8.
- [13] S. Xu, S. Yang, and F. C. M. Lau, "Keyword extraction and headline generation using novel word features." in *AAAI*, 2010, pp. 1461–1466.
- [14] L. C. Yu, H. Y. Lee, and L. S. Lee, "Abstractive headline generation for spoken content by attentive recurrent neural networks with asr error modeling," in *IEEE Workshop on Spoken Language Technology*, 2016.
- [15] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in *ICML*, vol. 14, 2014, pp. 1764–1772.
- [16] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.
- [17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
- [18] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.
- [19] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *ACL*, 2015.
- [20] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.
- [21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [22] K. Lopyrev, "Generating news headlines with recurrent neural networks," *arXiv preprint arXiv:1512.01712*, 2015.
- [23] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," *arXiv preprint arXiv:1603.07252*, 2016.
- [24] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals, "Sentence compression by deletion with lstms." in *EMNLP*, 2015, pp. 360–368.
- [25] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," *arXiv preprint arXiv:1603.06393*, 2016.
- [26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015.
- [27] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.
- [28] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*. IEEE, 2013, pp. 6645–6649.
- [29] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.
- [30] D. Graff and K. Chen, "Chinese gigaword ldc2003t09," *Linguistic Data Consortium*, 2003.
- [31] H. M. Wang, B. Chen, J. W. Kuo, S. S. Cheng *et al.*, "MATBN: A mandarin chinese broadcast news corpus," *International Journal of Computational Linguistics and Chinese Language Processing*, vol. 10, no. 2, pp. 219–236, 2005.
- [32] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8. Barcelona, Spain, 2004.
- [33] S. Chopra, M. Auli, A. M. Rush, and S. Harvard, "Abstractive sentence summarization with attentive recurrent neural networks," *Proceedings of NAACL-HLT*, pp. 93–98, 2016.
- [34] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.
- [35] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.
- [36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [37] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.
- [38] T.-H. K. Huang, Y.-N. Chen, and L. Kong, "Acbima: Advanced chinese bi-character word morphological analyzer," in *ACL-IJCNLP SIGHAN-8*, 2015.
- [39] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *Journal of the ACM (JACM)*, vol. 24, no. 4, pp. 664–675, 1977.