Cascaded Attention based Unsupervised Information Distillation for Compressive Summarization*

Piji Li[†] **Wai Lam**[†] **Lidong Bing**[‡] **Weiwei Guo**[§] **Hang Li**[♮]
[†]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
[‡]AI Lab, Tencent Inc., Shenzhen, China
[§]LinkedIn, Mountain View, CA, USA
[♮]Noah's Ark Lab, Huawei Technologies, Hong Kong
[†]{pjli, wlam}@se.cuhk.edu.hk, [‡]lyndonbing@tencent.com
[§]wguo@linkedin.com, [♮]hangli.hl@huawei.com

Abstract

When people recall and digest what they have read for writing summaries, the important content is more likely to attract their attention. Inspired by this observation, we propose a cascaded attention based unsupervised model to estimate the salience information from the text for compressive multi-document summarization. The attention weights are learned automatically by an unsupervised data reconstruction framework which can capture the sentence salience. By adding sparsity constraints on the number of output vectors, we can generate condensed information which can be treated as word salience. Fine-grained and coarse-grained sentence compression strategies are incorporated to produce compressive summaries. Experiments on some benchmark data sets show that our framework achieves better results than the state-of-the-art methods.

1 Introduction

The goal of Multi-Document Summarization (MDS) is to automatically produce a succinct summary, preserving the most important information of a set of documents describing a topic[1] (Luhn, 1958; Edmundson, 1969; Goldstein et al., 2000; Erkan and Radev, 2004b; Wan et al., 2007; Nenkova and McKeown, 2012). Considering the procedure of summary writing by humans, when people read, they will **remember** and **forget** part of the content. Information which is more important may make a deep impression easily. When people recall and digest what they have read to write summaries, the important information usually attracts more **attention** (*the behavioral and cognitive process of selectively concentrating on a discrete aspect of information, whether deemed subjective or objective, while ignoring other perceivable information*[2]) since it may repeatedly appears in some documents, or be positioned in the beginning paragraphs.

In the context of multi-document summarization, to generate a summary sentence for a key aspect of the topic, we need to find its relevant parts in the original documents, which may attract more attention. The semantic parts with high attention weights plausibly represent and reconstruct the topic's main idea. To this end, we propose a cascaded neural attention model to distill salient information from the input documents in an unsupervised data reconstruction manner, which includes two components: reader and recaller. The reader is a gated recurrent neural network (LSTM or GRU) based sentence sequence encoder which can map all the sentences of the topic into a global representation, with the mechanism of remembering and forgetting. The recaller decodes the global representation into significantly fewer diversified vectors for distillation and concentration. A cascaded attention mechanism is designed by incorporating attentions on both the hidden layer (dense distributed representation of a sentence) and the output layer (sparse bag-of-words representation of summary information). It is worth noting that the output vectors of the recaller can be viewed as word salience, and the attention matrix can be used as sentence salience. Both of them are automatically learned by data reconstruction in an **un-**

*The work described in this paper is supported by grants from the Research and Development Grant of Huawei Technologies Co. Ltd (YB2015100076/TH1510257) and the Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14203414).

[1]A topic represents a real event, e.g., "AlphaGo versus Lee Sedol".

[2]https://en.wikipedia.org/wiki/Attention (Apr., 2017)

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2071–2080
Copenhagen, Denmark, September 7–11, 2017.

supervised manner. Thereafter, the word salience is fed into a coarse-grained sentence compression component. Finally, the attention weights are integrated into a phrase-based optimization framework for compressive summary generation.

In fact, the notion of "attention" has gained popularity recently in neural network modeling, which has improved the performance of many tasks such as machine translation (Bahdanau et al., 2015; Luong et al., 2015). However, very few previous works employ attention mechanism to tackle MDS. Rush et al. (2015) and Nallapati et al. (2016) employed attention-based sequence-to-sequence (seq2seq) framework only for sentence summarization. Gu et al. (2016), Cheng and Lapata (2016), and Nallapati et al. (2016) also utilized seq2seq based framework with attention modeling for short text or single document summarization. Different from their works, our framework aims at conducting multi-document summarization in an unsupervised manner.

Our contributions are as follows: (1) We propose a cascaded attention model that captures salient information in different semantic representations. (2) The attention weights are learned automatically by an unsupervised data reconstruction framework which can capture the sentence salience. By adding sparsity constraints on the number of output vectors of the recaller, we can generate condensed vectors which can be treated as word salience; (3) We thoroughly investigate the performance of combining different attention architectures and cascaded structures. Experimental results on some benchmark data sets show that our framework achieves better performance than the state-of-the-art models.

2 Framework Description

2.1 Overview

Our framework has two phases, namely, information distillation for finding salient words/sentences, and compressive summary generation. For the first phase, our cascaded neural attention model consists of two components: reader and recaller as shown in Figure 1. The reader component reads in all the sentences in the document set corresponding to the topic/event. The information distillation happens in the recaller component where only the most important information is preserved. Precisely, the recaller outputs fewer vectors s than that of the input

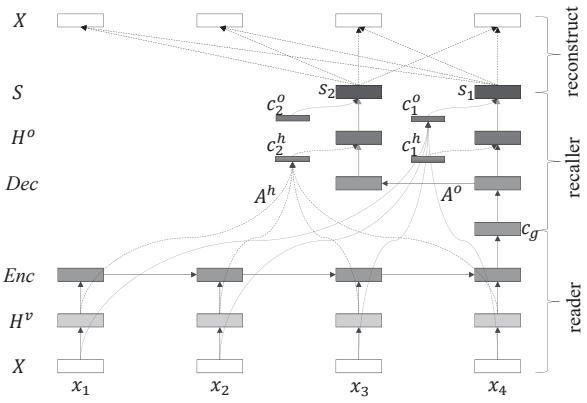

Figure 1: Our cascaded attention based unsupervised information distillation framework. X is the original input sentence sequence of a topic. H^i is the hidden vectors of sentences. "Enc" and "Dec" represent the RNN-based encoding and decoding layer respectively. c_g is the global representation for the whole topic. A^h and A^o are the distilled attention matrices for the hidden layer and the output layer respectively, representing the salience of sentences. H^o is the output hidden layer. s_1 and s_2 are the distilled condensed vectors representing the salience of words. Note that they are neither origin inputs nor golden summaries.

sentences x for the reader.

After the learning of the neural attention model finishes, the obtained salience information will be used in the second phase for compressive summary generation. This phase consists of two components: (i) the coarse-grained sentence compression component which can filter the trivial information based on the output vectors S from the neural attention model; (ii) the unified phrase-based optimization method for summary generation in which the attention matrix A^o is used to conduct fine-grained compression and summary construction.

2.2 Attention Modeling for Distillation

2.2.1 Reader

In the reader stage, for each topic, we extract all the sentences $X = \{x_1, x_2, \ldots, x_m\}$ from the set of input documents corresponding to a topic and generate a sentence sequence with length m. The sentence order is the same as the original order of the documents. Then the reader reads the whole sequence sentence by sentence. We employ the bag-of-words (BOW) representation as the initial semantic representation for sentences. Assume

that the dictionary size is k, then $x_i \in \mathbb{R}^k$.

Sparsity is one common problem for the BOW representation, especially when each vector is generated from a single sentence. Moreover, downstream algorithms might suffer from the curse of dimensionality. To solve these problems, we add a hidden layer H^v (v for input layer) which is a densely distributed representation above the input layer as shown in Figure 1. Such distributed representation can provide better generalization than BOW representation in many different tasks (Le and Mikolov, 2014; Mikolov et al., 2013). Specifically, the input hidden layer will project the input sentence vector x_j to a new space $\mathbb{R}^h$ according to Equation 1. Then we obtain a new sentence sequence $H^v = [h_1^v, h_2^v, \ldots, h_m^v]$.

$$h_j^v = \tanh(W_{xh}^v x_j + b_h^v) \tag{1}$$

where W_{xh}^v and b_h^v are the weight and bias respectively. The superscript v means that the variables are from the input layer.

While reading the sentence sequence, the reader should have the ability of remembering and forgetting. Therefore, we employ the RNN models with various gates (input gate, forget gate, etc.) to imitate the remembering and forgetting mechanism. Then the RNN based neural encoder (the third layer in Figure 1) will map the whole embedding sequence to a single vector c_g which can be regarded as a global representation for the whole topic. Let t be the index of the sequence state for the sentence x_t, the hidden unit h_t^e (e for encoder RNN) of the RNN encoder can be computed as:

$$h_t^e = f(h_{t-1}^e, h_t^v) \tag{2}$$

where the RNN $f(\cdot)$ computes the current hidden state given the previous hidden state h_{t-1}^e and the sentence embedding h_t^v. The encoder generates hidden states $\{h_t^e\}$ over all time steps. The last state $\{h_m^e\}$ is extracted as the global representation c_g for the whole topic. The structure for $f(\cdot)$ can be either an LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014).

2.2.2 Recaller

The recaller stage is a reverse of the reader stage, but it outputs less number of vectors in S as shown in Figure 1. Given the global representation c_g, the past hidden state h_{t-1}^d (d for decoder RNN) from the decoder layer, an RNN based decoder generates several hidden states according to:

$$h_t^d = f(h_{t-1}^d, c_g) \tag{3}$$

We use c_g to initialize the first decoder hidden state. The decoder will generate several hidden states $\{h_t^d\}$ over pre-defined time steps. Then, similar to the reader stage, we add an output hidden layer after the decoder layer:

$$h_t^o = \tanh(W_{hh}^o h_t^d + b_h^o) \tag{4}$$

where W_{hh}^o and b_h^o are the weight and bias respectively for the projection from h_t^d to h_t^o. Finally, the output layer maps these hidden vectors to the condensed vectors $S = [s_1, s_2, \ldots, s_n]$, Each output vector s_t has the same dimension k as the input BOW vectors and is obtained as follows:

$$s_t = \sigma(W_{hs} h_t^o + b_s) \tag{5}$$

For the purpose of distillation and concentration, we restrict n to be very small.

2.2.3 Cascaded Attention Modeling

Salience estimation for words and sentences is a crucial component in MDS, especially in the unsupervised summarization setting. We propose a cascaded attention model for information distillation to tackle the salience estimation task for MDS. We add attention mechanism not only in the hidden layer, but also in the output layer. By this cascaded attention model, we can capture the salience of sentences from two different and complementary vector spaces. One is the embedding space that provides better generalization, and the other one is the BOW vector space that captures more nuanced and subtle difference.

For each output hidden state h_t^o, we align it with each input hidden state h_i^v by an attention vector $a_{t,i}^h \in \mathbb{R}^m$ (recall that m is the number of input sentences). $a_{t,i}^h$ is derived by comparing h_t^o with each input sentence hidden state h_i^v:

$$a_{t,i}^h = \frac{\exp(score(h_t^o, h_i^v))}{\sum_{i'} \exp(score(h_t^o, h_{i'}^v))} \tag{6}$$

where $score(\cdot)$ is a content-based function to capture the relation between two vectors. Several different formulations can be used as the function $score(\cdot)$ which will be elaborated later.

Based on the alignment vectors $\{a_{t,i}^h\}$, we can create a context vector c_t^h by linearly blending the sentence hidden states $\{h_{i'}^v\}$:

$$c_t^h = \sum_{i'} a_{t,i'}^h h_{i'}^v \tag{7}$$

Then the output hidden state can be updated based on the context vector. Let $\tilde{h}_t^o = h_t^o$, then update the

original state according to the following operation:

$$h_t^o = \tanh(W_{ch}^a c_t^h + W_{hh}^a \tilde{h}_t^o) \quad (8)$$

The alignment vector $a_{t,i}^h$ captures which sentence should be attended more in the hidden space when generating the condensed representation for the whole topic.

Besides the attention mechanism on the hidden layer, we also directly add attention on the output BOW layer which can capture more nuanced and subtle difference information from the BOW vector space. The hidden attention vector $a_{t,i}^h$ is integrated with the output attention by a weight $\lambda_a \in [0, 1]$:

$$\bar{a}_{t,i}^o = \frac{\exp(score(s_t, x_i))}{\sum_{i'} \exp(score(s_t, x_{i'}))} \quad (9)$$

$$a_{t,i}^o = \lambda_a \bar{a}_{t,i}^o + (1 - \lambda_a) a_{t,i}^h \quad (10)$$

The output context vector is computed as:

$$c_t^o = \sum_{i'} a_{t,i'}^o x_{i'} \quad (11)$$

To update the output vector s_t in Equation 5, we develop a different method from that of the hidden attentions. Specifically we use a weighted combination of the context vectors and the original outputs with $\lambda_c \in [0, 1]$. Let $\tilde{s}_t = s_t$, then the updated s_t is:

$$s_t = \lambda_c c_t^o + (1 - \lambda_c)\tilde{s}_t \quad (12)$$

The parameters λ_a and λ_c can also be learned during training.

There are several different alternatives for the function $score(\cdot)$:

$$score(h_t, h_s) = \begin{cases} {h_t}^T h_s & dot \\ {h_t}^T W h_s & tensor \\ v^T \tanh(W[h_t; h_s]) & concat \end{cases} \quad (13)$$

Considering their behaviors as studied in (Luong et al., 2015), we adopt "*concat*" for the hidden attention layer, and "*dot*" for the output attention layer.

2.2.4 Unsupervised Learning

By minimizing the loss owing to using the condensed output vectors to reconstruct the original input sentence vectors, we are able to learn the solutions for all the parameters as follows.

$$\min_{\Theta} \frac{1}{2m} \sum_{i=1}^{m} \|x_i - \sum_{j=1}^{n} s_j a_{j,i}^o\|_2^2 + \lambda_s \|S\|_1 \quad (14)$$

where Θ denotes all the parameters in our model. In order to penalize the unimportant terms in the output vectors, we put a sparsity constraint on the rows of S using l_1-regularization, with the weight λ_s as a scaling constant for determining its relative importance.

Let $\bar{s}$ be the magnitude vector computed from the columns in S ($S \in \mathbb{R}^{n \times k}$). Once the training is finished, each dimension of the vector $\bar{s}$ can be regarded as the **word salience** score. According to Equation 14, $s_i \in S$ is used to reconstruct the original sentence space X, and $n \ll m$ (the number of sentences in X is much more than the number of vectors in S) Therefore a large value in $\bar{s}$ means that the corresponding word contains important information about this topic and it can serve as the word salience.

Moreover, the output layer attention matrix A^o can be regarded as containing the **sentence salience** information. Note that each output vector s_i is generated based on the cascaded attention mechanism. Assume that $a_i^o = A_{i,:}^o \in \mathbb{R}^m$ is the attention weight vector for s_i. According to Equation 9, a large value in a_i^o conveys a meaning that the corresponding sentence should contribute more when generating s_i. We also use the magnitude of the columns in A^o to represent the salience of sentences.

2.3 Compressive Summary Generation Phase

2.3.1 Coarse-grained Sentence Compression

Using the information distillation result from the cascaded neural attention model, we conduct coarse-grained compression for each individual sentence. Such strategy has been adopted in some multi-document summarization methods (Li et al., 2013; Wang et al., 2013; Yao et al., 2015). Our coarse-grained sentence compression jointly considers word salience obtained from the neural attention model and linguistically-motivated rules. The linguistically-motivated rules are designed based on the observed obvious evidence for uncritical information from the word level to the clause level, which include news headers such as "BEIJING, Nov. 24 (Xinhua) –", intra-sentential attribution such as ", police said Thursday", ", he said", etc. The information filtered by the rules will be processed according to the word salience score. Information with smaller salience score ($< \epsilon$) will be removed.

2.3.2 Phrase-based Optimization for Summary Construction

After coarse-grained compression on each single sentence as described above, we design a unified optimization method for summary generation. We refine the phrase-based summary construction model in (Bing et al., 2015) by adjusting the goal as compressive summarization. We consider the salience information obtained by our neural attention model and the compressed sentences in the coarse-grained compression component.

Based on the parsed constituency tree for each input sentence as described in Section 2.3.1, we extract the noun-phrases (NPs) and verb-phrases (VPs). The salience S_i of a phrase P_i is defined as:

$$S_i = \{\sum_{t \in P_i} tf(t) / \sum_{t \in Topic} tf(t)\} \times a_i \quad (15)$$

where a_i is the salience of the sentence containing P_i. $tf(t)$ is the frequency of the concept t (unigram/bigram) in the whole topic. Thus, S_i inherits the salience of its sentence, and also considers the importance of its concepts.

The overall objective function of our optimization formulation for selecting salient NPs and VPs is formulated as an integer linear programming (ILP) problem:

$$\max\{\sum_i \alpha_i S_i - \sum_{i<j} \alpha_{ij}(S_i + S_j)R_{ij}\} \quad (16)$$

where α_i is the selection indicator for the phrase P_i, S_i is the salience scores of P_i, α_{ij} and R_{ij} is the co-occurrence indicator and the similarity of a pair of phrases (P_i, P_j) respectively. The similarity is calculated by the Jaccard Index based method. Specifically, this objective maximizes the salience score of the selected phrases as indicated by the first term, and penalizes the selection of similar phrase pairs.

In order to obtain coherent summaries with good readability, we add some constraints into the ILP framework such as sentence generation constraint: Let β_k denote the selection indicator of the sentence x_k. If any phrase from x_k is selected, $\beta_k = 1$. Otherwise, $\beta_k = 0$. For generating a compressed summary sentence, it is required that if $\beta_k = 1$, at least one NP and at lease one VP of the sentence should be selected. It is expressed as:

$$\forall P_i \in x_k, \alpha_i \leq \beta_k \wedge \sum_i \alpha_i \geq \beta_k, \quad (17)$$

Other constraints include sentence number, summary length, phrase co-occurrence, etc. For details, please refer to McDonald (2007), Woodsend and Lapata (2012), and Bing et al. (2015).

The objective function and constraints are linear. Therefore the optimization can be solved by existing ILP solvers such as the simplex algorithm (Dantzig and Thapa, 2006). In the implementation, we use a package called lp_solve[3].

In the post-processing, the phrases and sentences in a summary are ordered according to their natural order if they come from the same document. Otherwise, they are ordered according to the timestamps of the corresponding documents.

3 Experimental Setup

3.1 Datasets

DUC: Both DUC 2006 and DUC 2007 are used in our evaluation. DUC 2006 and DUC 2007 contain 50 and 45 topics respectively. Each topic has 25 news documents and 4 model summaries. The length of the model summary is limited to 250 words. **TAC**: We also use TAC 2010 and TAC 2011 in our experiments. TAC 2011 is the latest standard summarization benchmark data set and it contains 44 topics. Each topic falls into one of 5 predefined event categories and contains 10 related news documents and 4 model summaries. TAC 2010 is used as the parameter tuning data set of our TAC evaluation.

3.2 Settings

For text processing, the input sentences are represented as BOW vectors with dimension k. The dictionary is created using unigrams and named entity terms. The word salience threshold ϵ used in sentence compression is 0.005. For the neural network framework, we set the hidden size as 500. All the neural matrix parameters $\mathcal{W}$ in hidden layers and RNN layers are initialized from a uniform distribution between $[-0.1, 0.1]$. Adadelta (Schmidhuber, 2015) is used for gradient based optimization. Gradient clipping is adopted by scaling gradients then the norm exceeded a threshold of 10. The maximum epoch number in the optimization procedure is 200. We limit the number of distilled vectors $n = 5$. The attention cascaded parameter λ_a and λ_c can be learned by our model. The sparsity penalty λ_s in Equation 14 is

[3] http://lpsolve.sourceforge.net/5.5/

Table 1: Comparisons on TAC 2010

System	R-1	R-2	R-SU4
CW	0.353	0.092	0.123
SC	0.346	0.083	0.116
AttenC-tensor-gru	0.339	0.078	0.115
AttenC-concat-gru	0.353	0.089	0.121
AttenC-dot-lstm	0.352	0.089	0.121
AttenH-dot-gru	0.348	0.086	0.119
AttenO-dot-gru	0.348	0.085	0.118
AttenC-dot-gru	**0.359**	**0.092**	**0.124**
(w\o coarse-comp)	0.351	0.089	0.122

0.001. Our neural network based framework is implemented using Theano (Bastien et al., 2012) on a single GPU of Tesla K80.

We use ROUGE score as our evaluation metric (Lin, 2004) with standard options. F-measures of ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) are reported.

4 Results and Discussions

4.1 Effect of Existing Salience Models and Different Attention Architectures

We quantitatively evaluate the performance of different variants on the dataset of TAC 2010. The experimental results are shown in Table 1. Note that the summary generation phase for different methods are the same, and only the salience estimation methods are different. Commonly used existing methods for salience estimation include: concept weight (**CW**) (Bing et al., 2015) and sparse coding (**SC**) (Li et al., 2015). As mentioned in Section 2.2.3, there are several alternatives for the attention scoring function $score(\cdot)$: **dot**, **tensor**, and **concat**. Moreover, we also design experiments to show the benefit of our cascaded attention mechanism versus the single attention method. **AttenC** denotes the cascaded attention mechanism. **AttenH** and **AttenO** represent the attention only on the hidden layer or the output layer respectively without cascaded combination.

Among all the methods, the cascaded attention model with *dot* structure achieves the best performance. The effect of different RNN models, such as LSTM and GRU, is similar. However, there are less parameters in GRU resulting in improvements for the efficiency of training. Therefore, we choose **AttenC-dot-gru** as the attention structure of our framework in the subsequent experiments. Moreover, the results without coarse-grained sentence compression (Section 2.3.1) show that the compression can indeed improve the sumamrization performance.

Table 2: Results on DUC 2006.

System	R-1	R-2	R-SU4
Random	0.280	0.046	0.088
Lead	0.308	0.048	0.087
LexRank	0.360	0.062	0.118
TextRank	0.373	0.066	0.125
MDS-Sparse	0.340	0.052	0.107
DSDR	0.377	0.073	0.117
RA-MDS	0.391	0.081	0.136
ABS-Phrase	0.392	0.082	0.137
C-Attention	**0.393***	**0.087***	**0.141***

Table 3: Results on DUC 2007.

System	R-1	R-2	R-SU4
Random	0.302	0.046	0.088
Lead	0.312	0.058	0.102
LexRank	0.378	0.075	0.130
TextRank	0.403	0.083	0.144
MDS-Sparse	0.353	0.055	0.112
DSDR	0.398	0.087	0.137
RA-MDS	0.408	0.097	0.150
ABS-Phrase	0.419	0.103	0.156
C-Attention	**0.423***	**0.107***	**0.161***

4.2 Main Results of Compressive MDS

We compare our system **C-Attention** with several unsupervised summarization baselines and state-of-the-art models. **Random** baseline selects sentences randomly for each topic. **Lead** baseline (Wasson, 1998) ranks the news chronologically and extracts the leading sentences one by one. **TextRank** (Mihalcea and Tarau, 2004) and **LexRank** (Erkan and Radev, 2004a) estimate sentence salience by applying the PageRank algorithm to the sentence graph. **PKUTM** (Li et al., 2011) employs manifold-ranking for sentence scoring and selection; **ABS-Phrase** (Bing et al., 2015) generates abstractive summaries using phrase-based optimization framework. Three other unsupervised methods based on sparse coding are also compared, namely, **DSDR** (He et al., 2012), **MDS-Sparse** (Liu et al., 2015), and **RA-MDS** (Li et al., 2015).

As shown in Table 2, Table 3, and Table 4, our system achieves the best results on all the ROUGE

Table 4: Results on TAC 2011.

System	R-1	R-2	R-SU4
Random	0.303	0.045	0.090
Lead	0.315	0.071	0.103
LexRank	0.313	0.060	0.102
TextRank	0.332	0.064	0.107
PKUTM	0.396	0.113	0.148
ABS-Phrase	0.393	0.117	0.148
RA-MDS	0.400	0.117	0.151
C-Attention	**0.400***	**0.121***	**0.153***

* Statistical significance tests show that our method is better than the best baselines.

metrics. The reasons are as follows: (1) The attention model can directly capture the salient sentences, which are obtained by minimizing the global data reconstruction error; (2) The cascaded structure of attentions can jointly consider the embedding vector space and bag-of-words vector space when conducting the estimation of sentence salience; (3) The coarse-grained sentence compression based on distilled word salience, and the fine-grained compression via phrase-based unified optimization framework can generate more concise and salient summaries. It is worth noting that PKUTM used a Wikipedia corpus for providing domain knowledge. The system **SWING** (Min et al., 2012) is the best system for TAC 2011. Our results are not as good as SWING. The reason is that SWING employs category-specific features and requires supervised training. These features help them select better category-specific content for the summary. In contrast, our model is basically **unsupervised**.

4.3 Linguistic Quality Evaluation

The linguistic quality of summaries generated by ABS-Phrase, PKUTM, and our model from 20 topics of TAC 2011 is evaluated using the five linguistic quality questions on grammaticality (Q1), non-redundancy (Q2), referential clarity (Q3), focus (Q4), and coherence (Q5) in Document Understanding Conferences (DUC). A Likert scale with five levels is employed with 5 being very good with 1 being very poor. A summary was blindly evaluated by three assessors on each question. The results are given in Table 5. PKUTM is an extractive method that picks the original sentences, hence it achieves higher score in Q1 grammaticality. ABS-Phrase is an abstractive method and can generate new sentences by merging different phrases, which decreases the grammaticality. Grammaticality of our compression-based framework is better than ABS-Phrase, but not as good as PKUTM. However, our framework performs the best on some other metrics such as Q2 (non-redundancy) and Q4 (focus). The reason is that our framework can compress and remove some uncritical and redundancy content from the original sentences, which leads to better performance on Q2 and Q4.

Table 5: Evaluation of linguistic quality.

System	Q1	Q2	Q3	Q4	Q5	AVG
ABS-Phrase	3.75	3.38	3.75	3.35	3.12	3.47
PKUTM	4.13	3.45	3.83	3.33	2.92	3.53
Ours	3.96	3.50	3.79	3.50	3.25	3.60

Table 6: Top-10 terms extracted from each topic according to the word salience

Topic 1	Topic 2	Topic 3
school	heart	HIV
shooting	disease	Africa
Auvinen	study	circumcision
Finland	risk	study
police	test	infection
video	blood	trial
Wednesday	red	woman
gunman	telomere	drug
post	level	health

4.4 Case Study: Distilled Word Salience

As mentioned above, the output vectors S in our neural model contain the distilled word salience information. In order to show the performance of word salience estimation, we select 3 topics (events) from different categories of TAC 2011: "Finland Shooting", "Heart Disease", and "Hiv Infection Africa". For each topic, we sort the dictionary terms according to their salience scores, and extract the top-10 terms as the salience estimation results as shown in Table 6. We can see that the top-10 terms reveal the most important information of each topic. For the topic "Finland Shooting", there is a sentence from the golden summary "A teenager at a school in Finland went on a shooting rampage Wednesday, November 11, 2007, killing 8 people, then himself." It is obvious that the top-10 terms from Table 6 can capture this main point.

4.5 Case Study: Attention-based Sentence Salience

In our model, the distilled attention matrix A^o can be treated as sentence salience estimation. Let $\widehat{a}$ be the magnitude of the columns in A^o and $\widehat{a} \in \mathbb{R}^m$. $\widehat{a}_i$ represents the salience of the sentence x_i. We collect all the attention vectors for 8 topics of TAC 2011, and display them as an image as shown in Figure 2. The x-axis represents the sentence id (we show at most 100 sentences), and the y-axis represents the topic id. The gray level of pixels in the image indicates different salience scores, where $dark$ represents a high salience score and $light$ represents a small score. Note that different topics seem to hold different ranges of salience scores because they have different number of sentences, i.e. m. According to Equation 9, topics containing more sentences will distribute the attention to more units, therefore, each sentence will get a relatively smaller attention weight. But this issue does not affect the performance of MDS since different topics are independently processed.

In Figure 2, there are some chunks in each topic (see Topic 3 as an example) having higher attention weights, which indeed automatically captures one characteristic of MDS: *sentence position is an important feature for news summarization*. As observed by several previous studies (Li et al., 2015; Min et al., 2012), the sentences in the beginning of a news document are usually more important and tend to be used for writing model summaries. Manual checking verified that those high-attention chunks correspond to the beginning sentences. Our model is able to automatically capture this information by assigning the latter sentences in each topic lower attention weights.

4.6 Summary Case Analysis

Table 7 shows the summary of the topic "*Hawkins Robert Van Maur*" in TAC 2011. The summary contains four sentences, which are all compressed with different compression ratio. Some uncritical information is excluded from the summary sentences, such as "*police said Thursday*" in S2, "*But*" in S3, and "*he said*" in S4. In addition, the VP "*killing eight people*" in S2 is also excluded since it is duplicate with the phrase "killed eight people" in S3. Moreover, from the case we can find that the compression operation did not harm the linguistic quality.

Table 7: The summary of the topic "*Hawkins Robert Van Maur*".

S1: The young gunman who opened fire at a mall busy with holiday shoppers appeared to choose his victims at random, according to police[~~, but a note he left behind hinted at a troubled life~~].
S2: The teenage gunman who went on a shooting rampage in a department store, [~~killing eight people,~~] may have smuggled an assault rifle into the mall underneath clothing[~~, police said Thursday~~].
S3: [~~But~~] police said it was Hawkins who went into an Omaha shopping mall on Wednesday and began a shooting rampage that killed eight people.
S4: Mall security officers noticed Hawkins briefly enter the Von Maur department store at Omaha's Westroads Mall earlier Wednesday[~~, he said~~].

5 Related Works

According to different machine learning paradigms, summarization models can be divided into supervised framework and unsupervised framework. Some previous works have been proposed based on unsupervised models. For example, Mihalcea and Tarau (2004) and Erkan and Radev (2004a) estimated sentence salience by applying the PageRank algorithm to the sentence graph. He et al. (2012), Liu et al. (2015), Li et al. (2015) and Song et al. (2017) employed sparse coding techniques for finding the salient sentences as summaries. Li et al. (2017) conducted salience estimation jointly considering reconstructions on several different vector spaces generated by a variational auto-ecoder framework.

Some recent works utilize attention modeling based recurrent neural networks to tackle the task of single-document summarization. Rush et al. (2015) proposed a sentence summarization framework based on a neural attention model using a supervised sequence-to-sequence neural machine translation model. Gu et al. (2016) combined a copying mechanism with the seq2seq framework to improve the quality of the generated summaries. Nallapati et al. (2016) also employed the typical attention modeling based seq2seq framework, but utilized a trick to control the vocabulary size to improve the training efficiency. However, few previous works employ attention mechanism to tackle the unsupervised MDS problem. In contrast, our attention-based framework can generate summaries for multi-document summarization

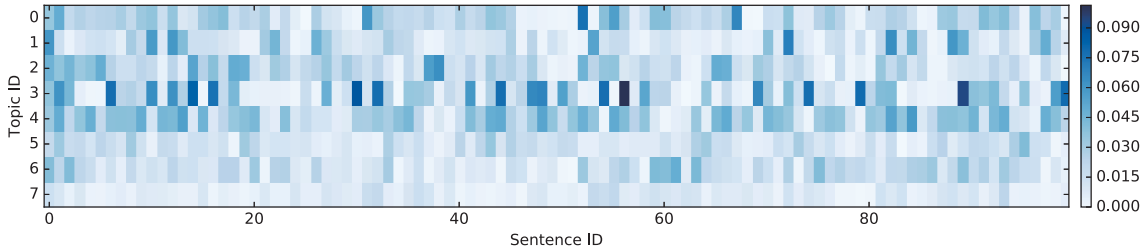

Figure 2: Visualization for sentence attention.

settings in an unsupervised manner.

6 Conclusions

We propose a cascaded neural attention based unsupervised salience estimation method for compressive multi-document summarization. The attention weights for sentences and salience values for words are both learned by data reconstruction in an unsupervised manner. We thoroughly investigate the performance of combining different attention architectures and cascaded structures. Experimental results on some benchmark data sets show that our framework achieves good performance compared with the state-of-the-art methods.

References

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

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *ACL*, pages 1587–1597.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *ACL*, pages 484–494.

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

George B Dantzig and Mukund N Thapa. 2006. *Linear programming 1: introduction*. Springer Science & Business Media.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Günes Erkan and Dragomir R Radev. 2004a. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP*, volume 4, pages 365–371.

Günes Erkan and Dragomir R Radev. 2004b. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, pages 457–479.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLPWorkshop*, pages 40–48.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, pages 1631–1640.

Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. 2012. Document summarization based on data reconstruction. In *AAAI*, pages 620–626.

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

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013. Document summarization via guided sentence compression. In *EMNLP*, pages 490–500.

Huiying Li, Yue Hu, Zeyuan Li, Xiaojun Wan, and Jianguo Xiao. 2011. PKUTM participation in TAC2011. In *TAC*.

Piji Li, Lidong Bing, Wai Lam, Hang Li, and Yi Liao. 2015. Reader-aware multi-document summarization via sparse coding. In *IJCAI*, pages 1270–1276.

Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 2017. Salience estimation via variational auto-encoders for multi-document summarization. In *AAAI*, pages 3497–3503.

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

He Liu, Hongliang Yu, and Zhi-Hong Deng. 2015. Multi-document summarization based on two-level sparse representation model. In *AAAI*, pages 196–202.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, pages 1412–1421.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *ECIR*, pages 557–564. Springer.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *EMNLP*, pages 404–411.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ziheng Lin Min, Yen Kan Chew, and Lim Tan. 2012. Exploiting category-specific information for multi-document summarization. *COLING*, pages 2093–2108.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer.

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

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Hongya Song, Zhaochun Ren, Shangsong Liang, Piji Li, Jun Ma, and Maarten de Rijke. 2017. Summarizing answers in non-factoid community question-answering. In *WSDM*, pages 405–414. ACM.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-ranking based topic-focused multi-document summarization. In *IJCAI*, volume 7, pages 2903–2908.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *ACL*, pages 1384–1394.

Mark Wasson. 1998. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *ACL*, pages 1364–1368.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *EMNLP-CNLL*, pages 233–243.

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2015. Compressive document summarization via sparse optimization. In *IJCAI*, pages 1376–1382.