Leveraging Contextual Sentence Relations for Extractive Summarization Using a Neural Attention Model

Pengjie Ren
jay.ren@outlook.com
Shandong University
Jinan, China

Zhumin Chen
chenzhumin@sdu.edu.cn
Shandong University
Jinan, China

Zhaochun Ren
renzhaochun@jd.com
Data Science Lab, JD.com
Beijing, China

Furu Wei
fuwei@microsoft.com
Microsoft Research Asia
Beijing, China

Jun Ma
majun@sdu.edu.cn
Shandong University
Jinan, China

Maarten de Rijke
derijke@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

As a framework for extractive summarization, sentence regression has achieved state-of-the-art performance in several widely-used practical systems. The most challenging task within the sentence regression framework is to identify discriminative features to encode a sentence into a feature vector. So far, sentence regression approaches have neglected to use features that capture contextual relations among sentences.

We propose a neural network model, Contextual Relation-based Summarization (CRSum), to take advantage of contextual relations among sentences so as to improve the performance of sentence regression. Specifically, we first use sentence relations with a word-level attentive pooling convolutional neural network to construct sentence representations. Then, we use contextual relations with a sentence-level attentive pooling recurrent neural network to construct context representations. Finally, CRSum automatically learns useful contextual features by jointly learning representations of sentences and similarity scores between a sentence and sentences in its context. Using a two-level attention mechanism, CRSum is able to pay attention to important content, i.e., words and sentences, in the surrounding context of a given sentence.

We carry out extensive experiments on six benchmark datasets. CRSum alone can achieve comparable performance with state-of-the-art approaches; when combined with a few basic surface features, it significantly outperforms the state-of-the-art in terms of multiple ROUGE metrics.

CCS CONCEPTS

•Information systems →Summarization;

ACM Reference format:
Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Jun Ma, and Maarten de Rijke. 2017. Leveraging Contextual Sentence Relations for Extractive Summarization Using a Neural Attention Model. In *Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, August 07-11, 2017,* 10 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080792

SIGIR '17, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080792

1 INTRODUCTION

Extractive summarization aims to generate a short text summary for a document or a set of documents by selecting salient sentences in the document(s) [35]. In recent years, sentence regression has emerged as an extractive summarization framework that achieves state-of-the-art performance [3, 50]; it has been widely used in practical systems [16, 18, 40, 49]. There are two major components in sentence regression: *sentence scoring* and *sentence selection*. The former scores a sentence to measure its importance, and the latter chooses sentences to generate a summary by considering both the importance scores and redundancy.

Sentence scoring has been extensively investigated in extractive summarization. Many approaches [3, 33] directly measure the salience of sentences whereas others [13, 23] first rank words (or bi-grams) and then combine these scores to rank sentences. Traditional scoring methods incorporate feature engineering as a necessary but labor-intensive task. To the best of our knowledge, most features of these methods are surface features, such as sentence length, sentence position, TF-IDF based features, etc. In Table 1, we list the scores achieved by *t-SR* [40], a traditional feature engineering-based sentence regression method for extractive summarization that achieves state-of-the-art performance. We also list an upper bound for the performance of sentence regression, which is obtained by scoring the sentences against human written summaries. There is a sizable gap in performance between *t-SR* and the upper bound. We believe that the reason for this is that none of *t-SR*'s features tries to encode semantic information.

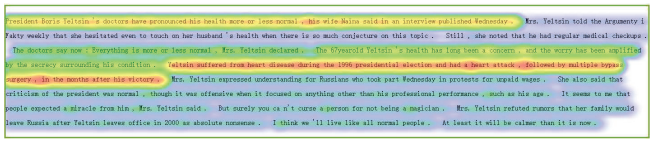
Recent neural network-based methods for *abstractive* summarization have addressed this matter [7, 31, 42]. Extracting semantic features via neural networks has received increased attention, also for extractive summarization [2, 3, 6]. Latent features learned by neural networks have been proven effective. PriorSum [3] is a recent example. To the best of our knowledge, PriorSum achieves the best performance on the three datasets listed in Table 1. But all methods, including PriorSum, extract latent features from standalone sentences. None considers their contextual relations.

Table 1: Multi-document summarization. ROUGE (%) of Sentence Regression (with greedy based sentence selection). Upper bounds are determined by scoring the sentences against human written summaries.

Dataset	Approach	ROUGE-1	ROUGE-2
DUC 2001	*t-SR*	34.82	7.76
	PriorSum	35.98	7.89
	Upper bound	40.82	14.76
DUC 2002	*t-SR*	37.33	8.98
	PriorSum	36.63	8.97
	Upper bound	43.78	15.97
DUC 2004	*t-SR*	37.74	9.60
	PriorSum	38.91	10.07
	Upper bound	41.75	13.73

(a) General-to-specific

(b) Specific-to-general



(c) Specific-to-general-to-specific

Figure 1: Sentence contexts in different instances from the DUC 2004 dataset. The color depth represents the importance of the sentence in terms of ROUGE-2 based on human written summaries. (Best viewed in color.)

We argue that sentence importance also depends on contextual relations, i.e., on relations of a sentence with its surrounding sentences. Figure 1(a) illustrates a general-to-specific paragraph structure, where the first sentence is a general summary of the event that is explained in detail by the following sentences. Figure 1(b) illustrates a specific-to-general paragraph structure, where the last sentence is a conclusion or reason of the event described by its preceding sentences. Figure 1(c) illustrates a specific-to-general-to-specific paragraph structure where the most important sentence is a connecting link between the preceding and the following context. So it summarizes both its preceding and following sentences.

We propose a hybrid neural model, namely Contextual Relation-based Summarization (CRSum), to automatically learn contextual relation features from data. CRSum applies a two-level attention mechanism (word-level and sentence-level) to attend differentially to more and less important content when constructing sentence/context representations. Specifically, we first leverage sentence relations using a convolutional neural network with word-level attentive pooling to construct sentence representations. Then, we leverage contextual relations using a recurrent neural network with sentence-level attentive pooling to construct context representations. With its two-level attention mechanism, CRSum can pay attention to more important content (words and sentences) in the surrounding context of a given sentence. Finally, CRSum jointly learns sentence/context representations as well as similarity scores between the sentence and its preceding/following context, which are regarded as the sentence's capacity to summarize its context.

We conduct extensive experiments on the DUC 2001, 2002, 2004 multi-document summarization datasets and the DUC 2005, 2006, 2007 query-focused multi-document summarization datasets. Our experimental results demonstrate that CRSum alone can achieve comparable performance to the state-of-the-art approaches. When combined with a few basic Surface Features (SF), CRSum+SF outperforms the state-of-the-art approaches in terms of ROUGE metrics.

To sum up, the main contributions in this paper are three-fold:

- We propose a neural model, CRSum, to take a sentence's contextual relations with its surrounding sentences into consideration for extractive summarization. CRSum jointly learns sentence and context representations as well as their similarity measurements. The measurements are used to estimate a sentence's ability to summarize its local context.
- We fuse contextual relations with a two-level attention mechanism in CRSum. With the mechanism, CRSum can learn to pay attention to important content (words and sentences) in the surrounding sentences of a given sentence.
- We carry out extensive experiments and analyses on six benchmark datasets. The results indicate that CRSum can significantly improve the performance of extractive summarization by modeling the contextual sentence relations.

2 RELATED WORK

We group related work on extractive summarization in three categories, which we discuss below.

2.1 Unsupervised techniques

In early studies on extractive summarization, sentences are scored by employing unsupervised techniques [37, 52]; centroid-based and Maximum Marginal Relevance (MMR)-based approaches are prominent examples. Centroid-based methods use sentence centrality as to indicate importance [29]. Radev et al. [37, 38] model cluster centroids in their summarization system, MEAD. LexRank (or TextRank) computes sentence importance based on eigenvector centrality in a graph of sentence similarities [10, 30]. Wan et al. [45–48] propose several centroid-based approaches for summarization. MMR-based methods consider a linear trade-off between relevance and redundancy [5]. Goldstein et al. [14] extend MMR to support extractive summarization by incorporating additional information about the document set and relations between the documents. McDonald [28] achieves good results by reformulating MMR as a knapsack packing problem and solving it using ILP. Later, Lin and Bilmes [26, 27] propose a variant of the MMR framework that maximizes an objective function that considers the linear trade-off between coverage and redundancy terms.

2.2 Feature engineering based techniques

Machine learning techniques have been used for better estimations of sentence importance. Kupiec et al. [22] train a Naive Bayes classifier to decide whether to include a sentence in the summary. Li et al. [24] evaluate sentence importance with support vector regression, after which a rule-based method is applied to remove redundant phrases. Gillick and Favre [13] evaluate bi-gram importance and use the scores to evaluate sentence importance and redundancy with a linear combination. Lin and Bilmes [26] propose a structural SVM learning approach to learn the weights of feature combinations using the MMR-like submodularity function proposed by Lin and Bilmes [26, 27]. Yan and Wan [51] propose the Deep Dependency Sub-Structure (DDSS) and topic-sensitive Multi-Task Learning (MTL) model. Given a document set, they parse all sentences into deep dependency structures with a Head-driven Phrase Structure Grammar parser and mine the frequent DDSSs after semantic normalization. They then employ MTL to learn the importance of these frequent DDSSs. Hu and Wan [19] propose PPSGen to automatically generate presentation slides by selecting and aligning key phrases and sentences. These methods all rely on human-engineered features. Most of the used features are surface features that do not take contextual relations into account.

2.3 Deep learning based techniques

Deep learning techniques have attracted considerable attention in the summarization literature, e.g., abstractive summarization [1, 7, 31], sentence summarization [11, 17, 42] and extractive summarization [2, 3, 6]. We focus on the use of deep learning techniques for extractive summarization. Kågebäck et al. [20] and Kobayashi et al. [21] use the sum of trained word embeddings to represent sentences or documents. They formalize the summarization task as the problem of maximizing a submodular function based on the similarities of the embeddings. Yin and Pei [53] propose CNNLM, a model based on convolutional neural networks, to project sentences into dense distributed representations, then model sentence redundancy by cosine similarity. Cao et al. [3] develop a summarization system called PriorSum, which applies enhanced convolutional neural networks to capture the summary prior features derived from length-variable phrases. In other work, the authors develop a ranking framework based on recursive neural networks (R2N2) to rank sentences for multi-document summarization. R2N2 formulates the ranking task as a hierarchical regression process that simultaneously measures the salience of a sentence and its constituents (e.g., phrases) in the parse tree [2]. Cheng and Lapata [6] treat single document summarization as a sequence labeling task and model it with recurrent neural networks. Their model is composed of a hierarchical document encoder and an attention-based extractor; the encoder derives the meaning representation of a document based on its sentences and their constituent words while the extractor adopts a variant of neural attention to extract sentences or words. Cao et al. [4] propose a system called AttSum for query-focused multi-document summarization that applies an attention mechanism to simulate the attentive reading of human behavior when a query is given.

A growing number of publications on extractive summarization focus on deep learning techniques. Unlike these publications, we propose a hybrid deep neural network that leverages the contextual information reflected by contextual sentence relations, which, to the best of our knowledge, are not considered in existing studies.

Table 2: Basic surface features used in this paper.

Feature	Description
$f_{len}(S_t)$	Length of S_t
$f_{pos}(S_t)$	Position of S_t in its document
$f_{tf}(S_t) = \frac{\sum_{w \in S_t} TF(w)}{f_{len}(S_t)}$	Average term frequency. $TF(w)$ is the term frequency of word w
$f_{df}(S_t) = \frac{\sum_{w \in S_t} DF(w)}{f_{len}(S_t)}$	Average document frequency. $DF(w)$ is the document frequency of word w

3 METHOD

3.1 Overview

There are two phases in our method to generate a summary: *sentence scoring* and *sentence selection*. In the sentence scoring phase, we learn a scoring function $f(S_t \mid \theta)$ for each sentence S_t to fit the ground truth ROUGE-2 score[1], i.e., ROUGE-2($S_t \mid S_{ref}$):

$$f(S_t \mid \theta) \sim \text{ROUGE-2}(S_t \mid S_{ref}) \tag{1}$$

where θ are the parameters; ROUGE-2($S_t \mid S_{ref}$) is the ground truth score of S_t in terms of ROUGE-2 based on human written summaries S_{ref} [25]. As with existing studies [3, 36, 40], we also use ROUGE-2 recall as the ground truth score. In §3.2, we detail how we model $f(S_t \mid \theta)$. During the sentence selection phase, we select a subset of sentences as the summary Ψ subject to a given length constraint l, i.e.,

$$\begin{aligned} \Psi^* &= \arg\max_{\Psi \subseteq D} \sum_{S_t \in \Psi} f(S_t \mid \theta) \\ &\text{such that } \sum_{S_t \in \Psi} |S_t| \leq l \text{ and } r(\Psi) \text{ hold,} \end{aligned} \tag{2}$$

where D is the set of sentences from one or more documents that belong to the same topic; $|S_t|$ is the length of S_t in words or bytes; $r(\Psi)$ is a constraint function to avoid redundancy in the final summary. Details of the algorithm are explained in §3.3.

3.2 Sentence scoring

Given a sentence S_t, we assume that its preceding context sentence sequence is $C_{pc} = \{S_{t-m}, \ldots, S_{t-c}, \ldots, S_{t-1} \mid 1 \leq c \leq m\}$ and that its following context sentence sequence is $C_{fc} = \{S_{t+1}, \ldots, S_{t+c}, \ldots, S_{t+n} \mid 1 \leq c \leq n\}$. Settings of m and n are discussed in §4 below. We use $f_{pc}(\mathrm{v}(S_t), \mathrm{v}_{pc}(S_t))$ to estimate the ability of S_t to summarize its preceding context:

$$f_{pc}(\mathrm{v}(S_t), \mathrm{v}_{pc}(S_t)) = \cos(\mathrm{v}(S_t), \mathrm{v}_{pc}(S_t)). \tag{3}$$

Similarly, $f_{fc}(\mathrm{v}(S_t), \mathrm{v}_{fc}(S_t))$ estimates the ability of S_t to summarize its following context:

$$f_{fc}(\mathrm{v}(S_t), \mathrm{v}_{fc}(S_t)) = \cos(\mathrm{v}(S_t), \mathrm{v}_{fc}(S_t)), \tag{4}$$

where $\mathrm{v}(S_t)$ is the sentence model of S_t; cos indicates the cosine similarity; $\mathrm{v}_{pc}(S_t)$ and $\mathrm{v}_{fc}(S_t)$ are the context models of C_{pc} and C_{fc}.

[1] http://www.berouge.com/Pages/default.aspx

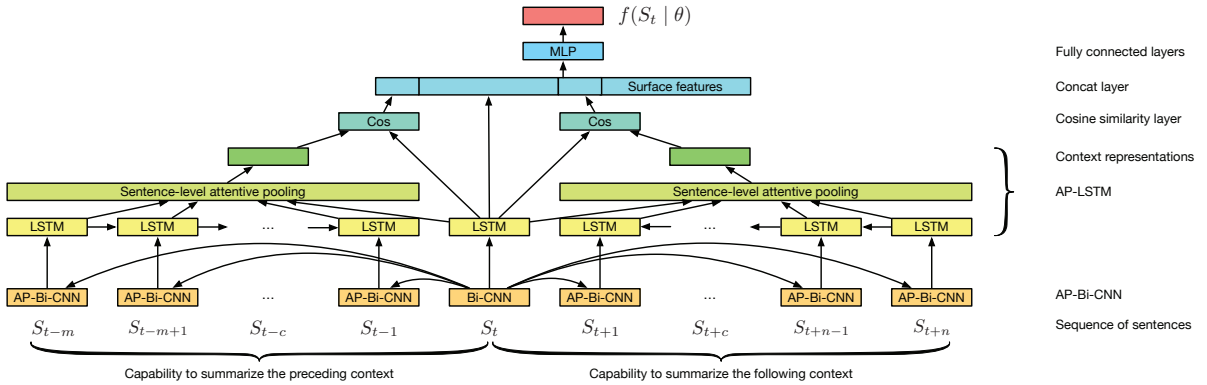


Figure 2: Architecture of CRSum+SF.

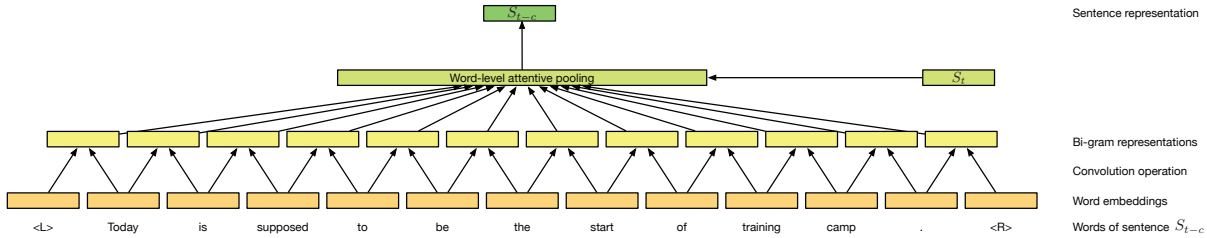


Figure 3: Attentive Pooling Bi-gram Convolutional Neural Network (AP-Bi-CNN) for sentence modeling.

Our model **CRSum+SF** is shown in Figure 2, where we combine $\mathrm{v}(S_t)$, $f_{pc}(\mathrm{v}(S_t), \mathrm{v}_{pc}(S_t))$ and $f_{fc}(\mathrm{v}(S_t), \mathrm{v}_{fc}(S_t))$ with four basic Surface Features (**SF**) listed in Table 2. Then we apply a MultiLayer Perceptron (MLP) [12, 41] as the decoder to transform the features into a single value as the final salience score to S_t, as shown in Eq. 5:

$$f(S_t \mid \theta) = \mathrm{MLP}\left(\begin{bmatrix} f_{pc}(\mathrm{v}(S_t), \mathrm{v}_{pc}(S_t)) \\ f_{fc}(\mathrm{v}(S_t), \mathrm{v}_{fc}(S_t)) \\ \mathrm{v}(S_t) \\ f_{len}(S_t) \\ f_{pos}(S_t) \\ f_{tf}(S_t) \\ f_{df}(S_t) \end{bmatrix}\right). \tag{5}$$

Here, θ are the parameters of the neural network. We use a 3 hidden layers MLP with tanh activation function; the sizes of the layers are 100, 50, and 1. Increasing the number and dimension size of layers has little influence on the performance according to our experiments. The model without the Surface Features (**SF**) in Table 2 is referred to as **CRSum**.

As with existing studies [2, 3, 40], we use the standard Mean Square Error (MSE) as the loss function to train CRSum (and CRSum+SF):

$$\begin{aligned} L(\theta) &= \frac{1}{|C| \cdot |D|} \sum_{D \in C} \sum_{S_t \in D} Err(S_t) \\ Err(S_t) &= \Big(f(S_t \mid \theta) - \text{ROUGE-2}(S_t \mid S_{ref})\Big)^2, \end{aligned} \tag{6}$$

where C is the set of all documents.

Sentence modeling: $\mathbf{v}(S_t)$. Since we conduct regression with respect to ROUGE-2, which is computed as the bi-gram overlap between the system generated summary and the human written summary, we use Bi-CNN [3] to model each sentence. We first concatenate adjacent words into bi-grams:

$$\mathrm{bi}(i, i+1) = \begin{bmatrix} \mathrm{v}_i \\ \mathrm{v}_{i+1} \end{bmatrix}, \tag{7}$$

where v_i is the word embedding for the i-th word of a sentence. After that, we perform convolutions on the bi-grams with a filter matrix:

$$\mathrm{v}_{bi}(i, i+1) = f(\mathrm{W}_c^T \cdot \mathrm{bi}(i, i+1) + b), \tag{8}$$

where $\mathrm{W}_c \in \mathbb{R}^{2|\mathrm{v}_i| \times |\mathrm{v}_i|}$ is the filter matrix; b is the bias; and $f(\cdot)$ is the activation function. We use the $\tanh(\cdot)$ function in our experiments.

Then we perform element-wise max pooling over the bi-gram representations $V_{bi}(S_t) = \{\mathrm{v}_{bi}(i, i+1) \mid 0 \le i \le |S_t|\}$ to get the sentence S_t's representation $\mathrm{v}(S_t)$:

$$\mathrm{v}(S_t) = \max_{\mathrm{v}_{bi}(i,i+1) \in V_{bi}(S_t)} \mathrm{v}_{bi}(i, i+1). \tag{9}$$

The function max chooses the maximum value of each dimension of the vectors in $V_{bi}(S_t)$.

In order to selectively encode the more important bi-grams into the sentence representations, an Attentive Pooling Convolutional Neural Network (AP-Bi-CNN) is applied, as shown in Figure 3. The difference with Bi-CNN is that we jointly learn a bi-gram weight $w_{bi}(i, i+1)$ when conducting pooling:

$$\mathrm{v}(S_{t-c}) = \max_{\mathrm{v}_{bi}(i,i+1) \in V_{bi}(S_{t-c})} w_{bi}(i, i+1) \cdot \mathrm{v}_{bi}(i, i+1). \tag{10}$$

Here, S_t is the sentence to conduct regression on; S_{t-c} is S_t's context sentence; and $w_{bi}(i, i+1)$ is the attention weight for the bigram vector $\mathrm{v}_{bi}(i, i+1)$.

Unlike existing attentive pooling techniques [8, 54], we use sentence relations to learn the pooling weights in Eq. 11:

$$\begin{bmatrix} w_{bi}(0,1) \\ \vdots \\ w_{bi}(i,i+1) \\ \vdots \\ w_{bi}(|S_{t-c}|,|S_{t-c}+1|) \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} \cos(\mathrm{v}_{bi}(0,1),\mathrm{v}(S_t)) \\ \vdots \\ \cos(\mathrm{v}_{bi}(i,i+1),\mathrm{v}(S_t)) \\ \vdots \\ \cos(\mathrm{v}_{bi}(|S_{t-c}|,|S_{t-c}+1|),\mathrm{v}(S_t)) \end{bmatrix}\right). \tag{11}$$

We use the softmax function to normalize the weights. The index of w_{bi} starts from 0 and ends with $|S_{t-c}+1|$ because we add two padding words "<L>" (Left) and "<R>" (Right) to each sentence, as shown in Figure 3.

Context modeling: $\mathbf{v}_{pc}(S_t)$ and $\mathbf{v}_{fc}(S_t)$. In order to model the relations between a sentence and its context, we also need to encode the context sentences into a vector representation. Recurrent Neural Networks with a Long Short-Term Memory (LSTM) unit have been successfully applied to many sequence modeling tasks [11, 31, 42]. There are many variations of LSTM that differ in their connectivity structure and activation functions. The LSTM architecture we use is given by the following equations [15]:

$$\begin{aligned} &\text{LSTM}: \mathrm{h}_{t-1}, \mathrm{v}(S_t), c_{t-1} \rightarrow c_t, \mathrm{h}_t \\ &\mathrm{x}_t = \begin{bmatrix} \mathrm{h}_{t-1} \\ \mathrm{v}(S_t) \end{bmatrix} \\ &f_i = \text{sigm}(\mathrm{W}_i^T \cdot \mathrm{x}_t + b_i); \\ &f_f = \text{sigm}(\mathrm{W}_f^T \cdot \mathrm{x}_t + b_f) \\ &f_o = \text{sigm}(\mathrm{W}_o^T \cdot \mathrm{x}_t + b_o); \\ &f_g = \tanh(\mathrm{W}_g^T \cdot \mathrm{x}_t + b_g) \\ &c_t = f_f \odot c_{t-1} + f_i \odot f_g; \\ &\mathrm{h}_t = f_o \odot \tanh(c_t), \end{aligned} \tag{12}$$

where $\mathrm{W}_i, \mathrm{W}_f, \mathrm{W}_o, \mathrm{W}_g \in \mathbb{R}^{2|\mathrm{v}(S_t)| \times |\mathrm{v}(S_t)|}$ are the parameter matrices; b_i, b_f, b_o, b_g are the bias parameters; h_t is the hidden state with respective to the t-th time step input $\mathrm{v}(S_t)$; c_t is the memory cell vector of the t-th time step; and sigm and tanh are applied elementwise. LSTM has a complicated dynamics that allows it to easily "memorize" information for an extended number of timesteps. The "long term" memory is stored in a vector of memory cells c_t. LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next time step.

Given a sentence S_t, we recurrently apply the LSTM unit to its preceding context sentence sequence C_{pc} and following context sentence sequence C_{fc}. For each timestamp t, S_t is fed into the LSTM unit and a corresponding vector representation h_t is generated. Then, we have $V_{pc} = \{\mathrm{h}_{t-m}, \ldots, \mathrm{h}_{t-1}\}$ for C_{pc} and $V_{fc} = \{\mathrm{h}_{t+1}, \ldots, \mathrm{h}_{t+n}\}$ for C_{fc}. Finally, we encode V_{pc} and V_{fc} into vector representations with sentence-level attentive pooling which can attend differentially to more and less important sentences, as shown in Figure 2. The formula for S_t's preceding context is

$$\mathrm{v}_{pc}(S_t) = \max_{\mathrm{h}_{t-i} \in V_{pc}} w_{t-i} \cdot \mathrm{h}_{t-i}, \tag{13}$$

where w_{t-i} is the attention weight for the hidden context state h_{t-i}. The formula for S_t's following context is similar.

Unlike most existing attention mechanisms where the last hidden state of an LSTM is used to learn the attention weights [7, 42], here we apply contextual sentence relations to model attention weights:

$$\begin{bmatrix} w_{t-m} \\ \vdots \\ w_{t-i} \\ \vdots \\ w_{t-1} \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} \cos(\mathrm{h}_{t-m}, \mathrm{h}_t) \\ \vdots \\ \cos(\mathrm{h}_{t-i}, \mathrm{h}_t) \\ \vdots \\ \cos(\mathrm{h}_{t-1}, \mathrm{h}_t) \end{bmatrix}\right). \tag{14}$$

3.3 Sentence selection

There are two branches of commonly used algorithms for sentence selection, namely Greedy and Integer Linear Programming (ILP). Greedy is a little less promising than ILP because it greedily maximizes a function which ILP exactly maximizes. However, it offers a nice trade-off between performance and computation cost. Besides, since the objective (Eq. 2) is submodular, maximizing it with Greedy has a mathematical guarantee on optimality [26, 27, 32]. Thus, we use Greedy as the sentence selection algorithm. The algorithm starts with the sentence of the highest score. In each step, a new sentence S_t is added to the summary Ψ if it satisfies the following two conditions:

(1) It has the highest score in the remaining sentences;
(2) $\frac{\textit{bi-gram-overlap}(S_t, \Psi)}{f_{len}(S_t)} \leq 1 - \lambda$, where $\textit{bi-gram-overlap}(S_t, \Psi)$ is the count of bi-gram overlap between sentence S_t and the current summary Ψ.

The algorithm terminates when the length constraint is reached. Settings of λ are discussed in §7.2 below.

4 EXPERIMENTAL SETUP

We list the datasets and metrics used in §4.1 and introduce the implementation details of our model in §4.2.

4.1 Datasets and evaluation metrics

For evaluation we use well-known corpora made available by the Document Understanding Conference (DUC).[2] The DUC 2001, 2002 and 2004 datasets are for multi-document summarization. The DUC 2005, 2006 and 2007 datasets are for query-focused multi-document summarization. The documents are from the news domain and grouped into thematic clusters. For each document cluster, we concatenate all articles and split them into sentences using the tool provided with the DUC 2003 dataset. We follow standard practice and train our models on two years of data and test on the third.

The ROUGE metrics are the official metrics of the DUC extractive summarization tasks [39]. We use the official ROUGE tool[3] [25] to evaluate the performance of the baselines as well as our approaches.

[2] http://duc.nist.gov/

[3] ROUGE-1.5.5 with options: -n 2 -m -u -c 95 -x -r 1000 -f A -p 0.5 -t 0.

The length constraint is "-l 100" for DUC 2001/2002, "-b 665" for DUC 2004 and "-l 250" for DUC 2005/2006/2007. We take ROUGE-2 recall as the main metric for comparison because Owczarzak et al. [36] show its effectiveness for evaluating automatic summarization systems. For significance testing we use a two-tailed paired Student's t-test with $p < 0.05$.

4.2 Implementation details

Stanford CoreNLP[4] is used to tokenize the sentences. The 50 dimensional GloVe[5] vectors are used to initialize the word embeddings. We replace a word that is not contained in the GloVe vocabulary as "<U>" (Unknown). The word embeddings are fine-tuned during training. Before feeding the word embeddings into the neural models, we perform the dropout operation that sets a random subset of its argument to zero. The dropout layer acts as a regularization method to reduce overfitting during training [44]. To learn the weights of our model, we apply the diagonal variant of AdaGrad [9] with mini-batches, whose size we set to 20. The parameters m and n that represent the number of context sentences are considered from 1 to 10. We found that there is no further improvement for $m, n > 5$, so we set $m, n = 5$ in our experiments. The best settings of the parameter λ are decided by presenting the ROUGE-2 performance with λ ranging from 0 to 0.9 with a step size of 0.05.

5 TWO EXAMPLES

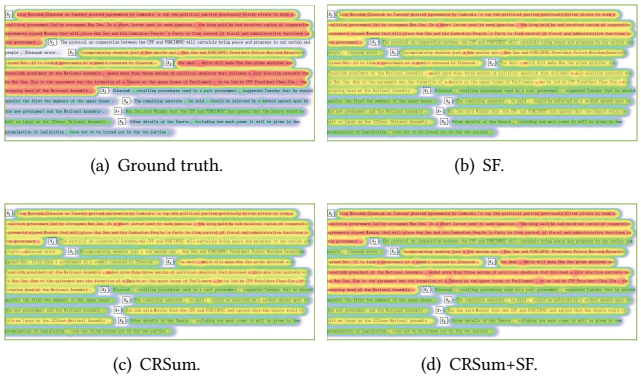
We present two examples to illustrate our methods at work. The first is an instance from the DUC 2004 dataset. From top-left to bottom-right, Figure 4 shows the ground truth, SF, CRSum, and CRSum+SF, respectively. The depth of the color corresponds to the importance of the sentence given by ground truth or models. We can see that, SF cannot significantly distinguish the different importance of different sentences. It wrongly estimates which of the two is more important, the third sentence or the fourth. CRSum is better than SF, however its capability is still limited in distinguishing different degrees of importance compared to the ground truth. In contrast, when combining CRSum and SF, CRSum+SF can better fit the ground truth.

As a second example, we visualize the learned two-level attentive pooling weights of an instance, as shown in Figure 5. Figure 5(a) illustrates word-level attentive pooling. We can see that S_t helps to pick up the more important words of S_{t+1} when modeling S_{t+1} into a vector representation. Figure 5(b) illustrates sentence-level attentive pooling. As shown, the context sentences S_{t+1} to S_{t+5} are treated differently according to their relevance to S_t. The more relevant sentences have more effect on the final results.

6 RESULTS

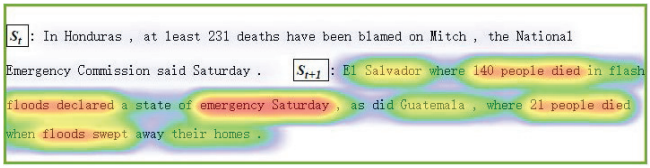
In §6.1, we compare CRSum with several state-of-the-art methods on the DUC 2001, 2002 and 2004 multi-document summarization datasets. We confirm that modeling contextual sentence relations significantly improves the performance of extractive summarization. In §6.2 we evaluate the effectiveness of contextual features on the DUC 2005, 2006 and 2007 query-focused multi-document summarization datasets. Here, we show that modeling contextual sentence relations is also useful for query-focused summarization. We follow with further analyses of the results in §7.

[4]http://stanfordnlp.github.io/CoreNLP/
[5]http://nlp.stanford.edu/projects/glove/

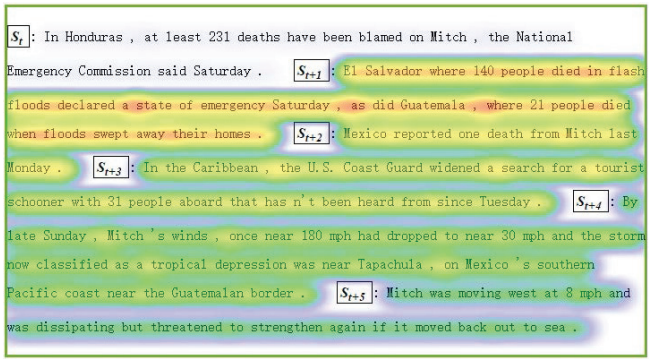


(a) Ground truth. (b) SF. (c) CRSum. (d) CRSum+SF.

Figure 4: Visualization of sentence scoring. The depth of the color corresponds to the importance of the sentence given by groundtruth or models. The boxed characters S_i indicate sentence start. (Best viewed in color.)



(a) Word-level attention. Color depth corresponds to the weight $w_{bi}(i, i+1)$ (Eq. 11).



(b) Sentence-level attention. Color depth corresponds to the weight $w_{t\text{-}i}$ (Eq. 14).

Figure 5: Visualization of word-level and sentence-level attention mechanisms. (Best viewed in color.)

6.1 Generic multi-document summarization

We first consider the generic multi-document summarization task. We list the methods compared against CRSum in Table 3. LexRank, ClusterHITS, ClusterCMRW are centroid-based methods; of these, ClusterHITS achieves the best ROUGE-1 score on DUC 2001. Lin is an MMR-based method. REGSUM, Ur, Sr, U+Sr and SF are feature engineering-based methods with different features. R2N2 uses an RNN to encode each sentence into a vector based on its parse tree, then performs sentence regression combined with 23 features. GA and ILP are greedy and ILP-based sentence selection algorithms,

Table 3: Methods considered for comparison in §6.1.

Acronym	Gloss	Reference
SF	Surface features with MLP as decoder	§3
CRSum	The proposed neural model in this paper	§3
CRSum (no attention)	CRSum without attention mechanism	§3
CRSum+SF	Combination of CRSum and SF	§3
Unsupervised methods		
LexRank	Centroid based method	[10]
ClusterHITS	Centroid based method	[48]
ClusterCMRW	Centroid based method	[48]
Lin	Maximal marginal relevance method	[27]
Feature engineering based methods		
REGSUM	Regression word saliency estimation	[2]
Ur	REGSUM with different features	[2]
Sr	SVR with 23 defined features	[2]
U+Sr	Combination of Ur and Sr	[2]
Deep learning based methods		
R2N2_GA	RNN with greedy sentence regression	[2]
R2N2_ILP	RNN with ILP sentence regression	[2]
PriorSum	REGSUM with different features	[3]

respectively. PriorSum uses a CNN to encode each sentence into a feature vector and then performs sentence regression combined with surface features.

The ROUGE scores of the methods listed in Table 3 on the DUC 2001, 2002 and 2004 datasets are presented in Table 4. For each metric, the best performance per dataset is indicated in bold face. Generally, CRSum+SF achieves the best performance in terms of both ROUGE-1 and ROUGE-2 on all three datasets. Although ClusterHITS achieves higher ROUGE-1 scores on DUC 2001, its ROUGE-2 scores are much lower. In contrast, CRSum+SF works quite stably across datasets. ClusterCMRW gets higher ROUGE-1 scores on DUC 2002 and its ROUGE-2 score is comparable with R2N2_GA, but CRSum+SF improves ClusterCMRW by over 1.6 percentage points (%pts) in terms of ROUGE-2.

The performance of CRSum is comparable to the state-of-the-art methods, R2N2_GA, R2N2_ILP and PriorSum. Note that CRSum is a pure neural network model while R2N2_GA, R2N2_ILP and PriorSum are combinations of neural models and dozens of hand-crafted features. The neural parts of R2N2_GA, R2N2_ILP and PriorSum model the standalone sentence, while CRSum further considers the local contextual relations.

When we combine CRSum with four basic surface features, there is a big improvement and CRSum+SF achieves the best performance. Specifically, CRSum+SF improves over PriorSum, the best method, in terms of ROUGE-2 by 1%pt on DUC 2001, 2002 and over 0.5%pt on DUC 2004. The improvements in terms of ROUGE-2 achieved on the three benchmark datasets are considered big [28, 39].

The main insight is that CRSum captures different factors than SF, which we will analyze in detail in §7.

6.2 Query-focused multi-document summarization

Next, we consider the performance of CRSum on the query-focused multi-document summarization task. We list the methods against which CRSum is compared in Table 6. LEAD simply selects the leading sentences to form a summary; it is often used as an official baseline of this task [4]. QUERY_SIM ranks sentences according to their TF-IDF cosine similarity to the query. MultiMR is a graph-based manifold ranking method. SVR and SF+QF are feature engineering-based methods. For the query-focused summarization task, the relevance of sentences to the query is an important feature, so we

Table 4: Multi-document summarization. ROUGE results (%) on DUC 2001, 2002, 2004 datasets. Per dataset, significant improvements over the underlined methods are marked with † (t-test, $p < .05$).

	Approach	ROUGE-1	ROUGE-2
DUC 2001	Peer T	33.03	7.86
	ClusterHITS*	**37.42**	6.81
	LexRank	33.43	6.09
	Ur*	34.28	6.66
	Sr*	34.06	6.65
	U+Sr*	33.98	6.54
	R2N2_GA*	35.88	7.64
	R2N2_ILP*	36.91	7.87
	PriorSum*	35.98	7.89
	SF	34.82	7.76
	CRSum	35.36	8.30
	CRSum+SF	36.54†	**8.75**†
DUC 2002	Peer 26	35.15	7.64
	ClusterCMRW*	38.55	8.65
	LexRank	35.29	7.54
	Ur*	34.16	7.66
	Sr*	34.23	7.81
	U+Sr*	35.13	8.02
	R2N2_GA*	36.84	8.52
	R2N2_ILP*	37.96	8.88
	PriorSum*	36.63	8.97
	SF	37.33	8.98
	CRSum	37.10	9.29
	CRSum+SF	**38.90**†	**10.28**†
DUC 2004	Peer 65	37.88	9.18
	REGSUM*	38.57	9.75
	LexRank	37.87	8.88
	Lin*	39.35	–
	Ur*	37.22	9.15
	Sr*	36.72	9.10
	U+Sr*	37.62	9.31
	R2N2_GA*	38.16	9.52
	R2N2_ILP*	38.78	9.86
	PriorSum*	38.91	10.07
	SF	37.74	9.60
	CRSum	38.19	9.66
	CRSum+SF	**39.53**†	**10.60**†

(Peer T, 26, 65 are the best performing participants at DUC 2001, 2002, 2004, respectively. Scores of the methods marked with * are taken from the corresponding references listed in Table 3.)

Table 5: Query Features (QF) used in this paper.

Feature	Description
$f_{q1}(S_t, S_q) = \cos(TF(S_t), TF(S_q))$	Cosine of TF vectors of sentence S_t and query S_q
$f_{q2}(S_t, S_q) = \cos(emb(S_t), emb(S_q))$	Cosine of average embedding vectors of S_t and S_q
$f_{q3}(S_t, S_q) = \frac{overlap(S_t, S_q)}{f_{len}(S_t)}$	Unigram overlap with respect to S_t
$f_{q4}(S_t, S_q) = \frac{overlap(S_t, S_q)}{f_{len}(S_q)}$	Unigram overlap with respect to S_q

Table 6: Methods considered for comparison in §6.2.

Acronym	Gloss	Reference
SF+QF	Surface features (Table 2+5) with MLP as decoder	§3
CRSum	The proposed neural model in this paper	§3
CRSum (no attention)	CRSum without attention mechanism	§3
CRSum+SF+QF	Combination of CRSum and SF	§3
Unsupervised methods		
LEAD	Select the leading sentences	[4]
QUERY_SIM	TF-IDF cosine similarity	[4]
MultiMR	Graph based manifold ranking method	[47]
Feature engineering based methods		
SVR	SVR with hand-crafted features	[34]
Deep learning based methods		
ISOLATION	Embedding and TF-IDF cosine similarity	[4]
DocEmb	Embedding distributions based summarization	[21]
AttSum	Neural attention summarization	[4]

combine the Surface Features (SF) in Table 2 and the Query Features (QF) in Table 5, using SF+QF to refer to the resulting method. ISOLATION contains two parts; sentence saliency is modeled as the cosine similarity between a sentence embedding and the document embedding; query relevance is modeled as the TF-IDF cosine similarity between a sentence and the query. DocEmb summarizes by asymptotically estimating KL-divergence based on document embedding distributions. AttSum learns distributed representations for sentences and the documents; it applies an attention mechanism to simulate human reading behavior.

The results on the query-focused multi-document summarization task on the DUC 2005, 2006 and 2007 datasets are presented in Table 7. Generally, CRSum alone is not enough for this task; it is outperformed by SF+QF. This is because CRSum does not consider relevance of a sentence given the queries; the relevance relation is captured by the features encoded in SF+QF. CRSum+SF+QF improves over SF+QF by 0.5%pt to 0.7%pt, which means that CRSum is also useful as a supplementary feature for the query-focused summarization task.

7 ANALYSIS

Having answered our main research questions in the previous section, we now analyze our experimental results and the impact of our modeling choices. We analyze the effectiveness of the learned contextual features compared to the surface features; we explore different settings of the threshold parameter λ in the greedy algorithm (sentence selection phase, §3.3) to determine the sensitivity of our method; and we analyze our attention mechanisms.

Table 7: Query-focused multi-document summarization. ROUGE results (%) on DUC 2005, 2006, 2007 datasets. Per dataset, significant improvements over the underlined methods are marked with † (t-test, p < .05).

	System	ROUGE-1	ROUGE-2
DUC 2005	Peer 15	37.52	7.25
	LEAD*	29.71	4.69
	QUERY_SIM*	32.95	5.91
	SVR*	36.91	7.04
	MultiMR*	35.58	6.81
	DocEmb*	30.59	4.69
	ISOLATION*	35.72	6.79
	AttSum*	37.01	6.99
	SF+QF	39.18	7.79
	CRSum	36.96	7.01
	CRSum+SF+QF	**39.52**†	**8.41**†
DUC 2006	Peer 24	41.11	9.56
	LEAD*	32.61	5.71
	QUERY_SIM*	35.52	7.10
	SVR*	39.24	8.87
	MultiMR*	38.57	7.75
	DocEmb*	32.77	5.61
	ISOLATION*	40.58	8.96
	AttSum*	40.90	9.40
	SF+QF	41.45	9.57
	CRSum	39.51	9.19
	CRSum+SF+QF	**41.70**	**10.03**†
DUC 2007	Peer 15	44.51	12.45
	LEAD*	36.14	8.12
	QUERY_SIM*	36.32	7.94
	SVR*	43.42	11.10
	MultiMR*	41.59	9.34
	DocEmb*	33.88	6.46
	ISOLATION*	42.76	10.79
	AttSum*	43.92	11.55
	SF+QF	44.29	11.73
	CRSum	41.20	11.17
	CRSum+SF+QF	**44.60**†	**12.48**†

(Peer 15, 24, 15 are the best performing participants at DUC 2005, 2006, 2007, respectively. Scores of the methods marked with * are taken from the corresponding references listed in Table 6.)

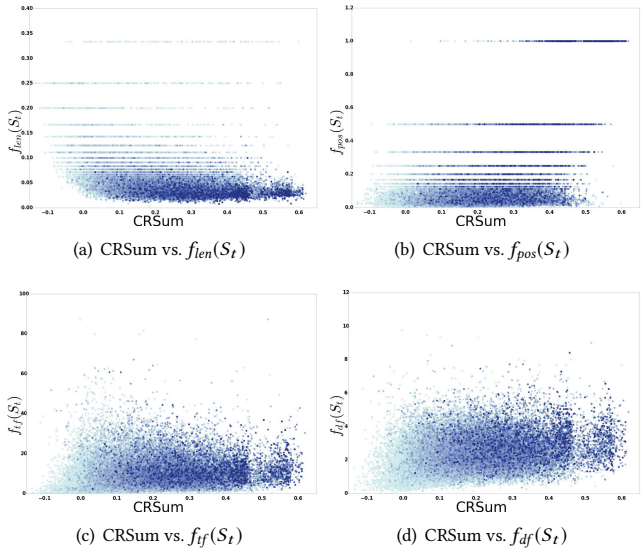
7.1 CRSum vs. the surface features

Pearson correlation coefficients can reflect the effectiveness of the feature to some extent. We examine correlations with the ground truth of the surface features in Table 2 and of CRSum, as shown in Table 8. CRSum achieves higher correlation scores than the surface features ($f_{len}(S_t)$, $f_{pos}(S_t)$, $f_{tf}(S_t)$ and $f_{df}(S_t)$) and comparable correlation scores with SF. The results also confirm that $f_{len}(S_t)$ and $f_{pos}(S_t)$ are important features for extractive summarization [2, 3, 49].

Pearson correlation coefficients only reflect linear correlations. Hence, we further visualize the relation between the feature space

Table 8: Pearson correlation coefficients of surface features and CRSum.

Features	DUC 2001	DUC 2002	DUC 2004
$f_{len}(S_t)$	0.31	0.31	0.37
$f_{pos}(S_t)$	0.28	0.31	0.40
$f_{tf}(S_t)$	0.14	0.20	0.24
$f_{df}(S_t)$	0.19	0.23	0.38
SF	0.45	**0.48**	**0.64**
CRSum	**0.46**	0.44	0.56



(a) CRSum vs. $f_{len}(S_t)$ (b) CRSum vs. $f_{pos}(S_t)$

(c) CRSum vs. $f_{tf}(S_t)$ (d) CRSum vs. $f_{df}(S_t)$

Figure 6: CRSum scores vs. surface feature scores. Each point represents a sentence. The color depth reflects the importance of the sentence according to the ground truth. (Best viewed in color.)

of CRSum and the surface features $f_{len}(S_t)$, $f_{pos}(S_t)$, $f_{tf}(S_t)$, and $f_{df}(S_t)$, as shown in Figure 6, by plotting CRSum scores against the feature values.[6] The color depth reflects the importance of a sentence according to the ground truth. Low CRSum scores mostly correspond to sentences with low ROUGE-2 scores, which means that CRSum can distinguish useless sentences effectively. Also, high CRSum scores mostly correspond to sentences with high ROUGE-2 scores, which means that CRSum can distinguish the most important sentences effectively. Obviously, this ability to identify the most important sentences is extremely useful, as a summary is usually short, containing just a few very important sentences; we should also note that this ability is still limited as there are low scoring and high scoring sentences mixed together.

7.2 Threshold parameter λ

Recall from §3.3 that after giving a salience score to each sentence, we greedily select the sentence with the highest score for inclusion in the final summary until the length constraint is reached. During the process, a parameter λ is used to avoid redundant sentences by discarding sentences whose bigram overlap with already selected

[6]The scores for CRSum range from −1 to 1 as its activation function is Tanh.

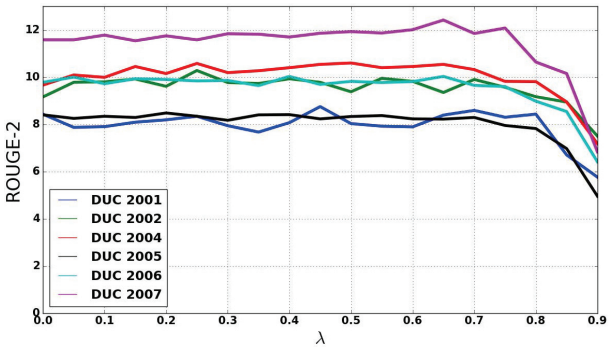


Figure 7: Sensitivity to the parameter λ of CRSum+SF during sentence selection.

sentences is larger than $1 - \lambda$. To investigate the sensitivity of our choice of λ, we examine the performance of CRSum+SF with the threshold parameter λ ranging from 0 to 0.9 with a step size of 0.05. The results are shown in Figure 7, where we plot performance in terms of ROUGE-2 against λ. Generally, the performance of CRSum+SF is not sensitive to the setting of λ for values less than 0.8, with the best performance achieved around 0.65 to 0.75.

7.3 Attention

We have illustrated our word and sentence level attention mechanisms with two examples in Figure 5. Here, we analyze the performance of these attention mechanisms in Table 9, using the same data as in §6.1. Generally, with two-level attentive pooling, CRSum gains around 0.3%pt–0.5%pt improvements in terms of ROUGE-2 over CRSum (without attention). The improvements are different on the three datasets with higher improvement on DUC 2001 and smaller improvement on DUC 2004. Interestingly, world-level and sentence-level attention yield comparable improvements over CRSum without attention. But their contribution is complementary as the combined attention mechanisms bring further improvements, on all metrics and datasets, demonstrating the need for both.

8 CONCLUSIONS & FUTURE WORK

This paper presents a novel neural network model, CRSum, to automatically learn features contained in sentences and in contextual relations between sentences. We have conducted extensive experiments on the DUC multi-document summarization datasets and query-focused multi-document summarization datasets. Without hand-crafted features, CRSum achieves a comparable performance with state-of-the-art methods. When combined with a few basic surface features, CRSum+SF significantly outperforms the baselines

Table 9: Analyzing attention mechanisms on the multi-document summarization task (%).

CRSum attention	DUC 2001 ROUGE-1/2	DUC 2002 ROUGE-1/2	DUC 2004 ROUGE-1/2
without	34.33/7.81	36.18/8.91	37.76/9.32
word-level	34.54/7.95	36.31/9.09	37.80/9.37
sentence-level	34.57/8.01	36.29/9.07	37.82/9.41
two-level	**35.36/8.30**	**37.10/9.29**	**38.19/9.66**

and achieves the best published performance on the DUC 2001, 2002 and 2004 datasets. Based on our experimental results and subsequent analyses, we conclude that CRSum encodes supplementary information that surface features cannot capture.

We believe our work can be advanced and extended in several directions: CRSum can be enriched by introducing a mechanism to explicitly model fine-grained sentence relations, such as parallelism relations, progressive relations, inductive reasoning relations and deductive reasoning relations [43]. Variants of CRSum can be also extended to other tasks, such as abstractive summarization and sentence summarization.

9 ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (61672322, 61672324), the Natural Science Foundation of Shandong province (2016ZRE27468), the Fundamental Research Funds of Shandong University, Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Dutch national program COMMIT, Elsevier, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] L. Bing, P. Li, Y. Liao, W. Lam, W. Guo, and R. J. Passonneau. Abstractive multi-document summarization via phrase selection and merging. In *ACL*, 2015.
[2] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou. Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI*, 2015.
[3] Z. Cao, F. Wei, S. Li, W. Li, M. Zhou, and H. Wang. Learning summary prior representation for extractive summarization. In *ACL*, 2015.
[4] Z. Cao, W. Li, S. Li, and F. Wei. Attsum: Joint learning of focusing and summarization with neural attention. In *COLING*, 2016.
[5] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
[6] J. Cheng and M. Lapata. Neural summarization by extracting sentences and words. In *ACL*, 2016.
[7] S. Chopra, M. Auli, and A. M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL-HLT*, 2016.
[8] C. N. dos Santos, M. Tan, B. Xiang, and B. Zhou. Attentive pooling networks. *CoRR*, 2016.
[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.
[10] G. Erkan and D. R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22(1):457–479, 2004.
[11] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals. Sentence compression by deletion with LSTMs. In *EMNLP*, 2015.
[12] M. W. Gardner and S. Dorling. Artificial neural networks (the multilayer perceptron) – A review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14):2627–2636, 1998.
[13] D. Gillick and B. Favre. A scalable global model for summarization. In *ILP-NLP*, 2009.
[14] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP*, 2000.
[15] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.
[16] K. Hong and A. Nenkova. Improving the estimation of word importance for news multi-document summarization. In *EACL*, 2014.
[17] B. Hu, Q. Chen, and F. Zhu. LCSTS: A large scale Chinese short text summarization dataset. In *EMNLP*, 2015.
[18] Y. Hu and X. Wan. PPSGen: Learning to generate presentation slides for academic papers. In *IJCAI*, 2013.
[19] Y. Hu and X. Wan. PPSGen: Learning-based presentation slides generation for academic papers. *TKDE*, 27(4):1085–1097, 2015.
[20] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi. Extractive summarization using continuous vector space models. In *CVSC@ EACL*, 2014.
[21] H. Kobayashi, M. Noguchi, and T. Yatsuka. Summarization based on embedding distributions. In *EMNLP*, 2015.
[22] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *SIGIR*, 1995.
[23] C. Li, X. Qian, and Y. Liu. Using supervised bigram-based ILP for extractive summarization. In *ACL*, 2013.
[24] S. Li, Y. Ouyang, W. Wang, and B. Sun. Multi-document summarization using support vector regression. In *DUC*, 2007.
[25] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL*, 2004.
[26] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *NAACL-HLT*, 2010.
[27] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *NAACL-HLT*, 2011.
[28] R. McDonald. A study of global inference algorithms in multi-document summarization. In *ECIR*, 2007.
[29] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *ACL*, 2004.
[30] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *EMNLP*, 2004.
[31] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *CoNLL*, 2016.
[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *MATH PROGRAM*, 14(1):265–294, 1978.
[33] Y. Ouyang, S. Li, and W. Li. Developing learning strategies for topic-based summarization. In *CIKM*, 2007.
[34] Y. Ouyang, W. Li, S. Li, and Q. Lu. Applying regression models to query-focused multi-document summarization. *IPM*, 47(2):227–237, 2011.
[35] P. Over and J. Yen. Introduction to DUC-2001: An intrinsic evaluation of generic news text summarization systems. In *DUC*, 2004.
[36] K. Owczarzak, J. M. Conroy, H. T. Dang, and A. Nenkova. An assessment of the accuracy of automatic evaluation in summarization. In *NAACL-HLT*, 2012.
[37] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP*, 2000.
[38] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *IPM*, 40(6):919–938, 2004.
[39] P. A. Rankel, J. M. Conroy, H. T. Dang, and A. Nenkova. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *ACL*, 2013.
[40] P. Ren, F. Wei, Z. Chen, J. Ma, and M. Zhou. A redundancy-aware sentence regression framework for extractive summarization. In *COLING*, 2016.
[41] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *TNN*, 1(4):296–298, 1990.
[42] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.
[43] W. Song, T. Liu, R. Fu, L. Liu, H. Wang, and T. Liu. Learning to identify sentence parallelism in student essays. In *COLING*, 2016.
[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15 (1):1929–1958, 2014.
[45] X. Wan. An exploration of document impact on graph-based multi-document summarization. In *EMNLP*, 2008.
[46] X. Wan. Using bilingual information for cross-language document summarization. In *NAACL-HLT*, 2011.
[47] X. Wan and J. Xiao. Graph-based multi-modality learning for topic-focused multi-document summarization. In *IJCAI*, 2009.
[48] X. Wan and J. Yang. Multi-document summarization using cluster-based link analysis. In *SIGIR*, 2008.
[49] X. Wan and J. Zhang. CTSUM: extracting more certain summaries for news articles. In *SIGIR*, 2014.
[50] X. Wan, Z. Cao, F. Wei, S. Li, and M. Zhou. Multi-document summarization via discriminative summary reranking. *CoRR*, 2015.
[51] S. Yan and X. Wan. Deep dependency substructure-based learning for multidocument summarization. *ACM Transactions on Information Systems (TOIS)*, 34(1):3, 2015.
[52] C. yew Lin and E. Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *ACL*, 2002.
[53] W. Yin and Y. Pei. Optimizing sentence modeling and selection for document summarization. In *IJCAI*, 2015.
[54] W. Yin, H. Schtze, B. Xiang, and B. Zhou. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4(1):259–272, 2016.