Neural Extractive Summarization with Side Information

Shashi Narayan   Nikos Papasarantopoulos   Shay B. Cohen   Mirella Lapata

Institute for Language, Cognition and Computation, School of Informatics, University of Edinburgh

{shashi.narayan,nikos.papasa}@ed.ac.uk {scohen,mlap}@inf.ed.ac.uk

Abstract

Most extractive summarization methods focus on the main body of the document from which sentences need to be extracted. However, the gist of the document may lie in side information, such as the title and image captions which are often available for newswire articles. We propose to explore side information in the context of single-document extractive summarization. We develop a framework for single-document summarization composed of a hierarchical document encoder and an attention-based extractor with attention over side information. We evaluate our model on a large scale news dataset. We show that extractive summarization with side information consistently outperforms its counterpart that does not use any side information, in terms of both informativeness and fluency.

1 Introduction

Increased access to information and the massive growth in the global news data have led to a growing demand from readers to spot emerging trends, person mentions and the evolution of storylines in the news (Liepins et al. 2017). The vast majority of this news data contains textual documents, driving the need for automatic document summarization systems aiming at acquiring key points in the form of a short summary from one or more documents.

While it is not so challenging for humans to summarize text, automatic summarization systems struggle with producing high quality summaries. Both extractive and abstractive systems have been proposed in recent years. Extractive summarization systems select sentences from the document and assemble them together to often generate a grammatical, fluent and semantically correct summary (Cheng and Lapata 2016; Nallapati, Zhai, and Zhou 2017; Yasunaga et al. 2017). Abstractive summarization systems, on the other hand, aim at building an internal semantic representation and then generate a summary from scratch (Chen et al. 2016; Nallapati et al. 2016; See, Liu, and Manning 2017; Tan and Wan 2017). Despite recent improvements, abstractive systems still struggle to outperform extractive systems. This paper addresses the task of single-document summarization and explores how we can further improve the sentence selection process for extractive summarization.

Most extractive methods often focus on the main body of the document from which sentences are extracted. Traditional methods manually define features which are local in the context of each sentence or a set of sentences which form the body of the document. Such features include sentence position and length (Radev et al. 2004), keywords and the presence of proper nouns (Kupiec, Pedersen, and Chen 1995; Mani 2001), frequency information such as content word frequency, composition functions for estimating sentence importance from word frequency, and the adjustment of frequency weights based on context (Nenkova, Vanderwende, and McKeown 2006) and low-level event-based features describing relationships between important actors in a document (Filatova and Hatzivassiloglou 2004). Sentences are ranked for extraction based on the overlap with features. Recent deep learning methods circumvent human-engineered features using continuous sentence features. Kågebäck et al. (2014) and Yin and Pei (2015) map sentences to a continuous vector space which is used for similarity measurement to reduce the redundancy in the generated summaries. Cheng and Lapata (2016) and Nallapati, Zhai, and Zhou (2017) use recurrent neural networks to read sequences of sentences to get a document representation which they use to label each sentence for extraction. These methods report state of the art results without using any kind of linguistic annotation.

It is a challenging task to rely only on the main body of the document for extraction cues, as it requires document understanding. Documents in practice often have side information, such as the title, image captions, videos, images and twitter handles, along with the main body of the document. These types of side information are often available for newswire articles. Figure 1 shows an example of a newswire article taken from CNN (CNN.com). It shows the side information such as the title (first block) and the images with their captions (third block) along with the main body of the document (second block). The last block shows a manually written summary of the document in terms of "highlights" to allow readers to quickly gather information on stories. As one can see in this example, gold highlights focus on sentences from the fourth paragraph, i.e., on key events such as the "PM's resignation", "bribery scandal and its investigation", "suicide" and "leaving an important note". Interestingly, the essence of the article is explicitly or implicitly

South Korean Prime Minister Lee Wan-koo offers to resign

Seoul (CNN) South Korea's Prime Minister Lee Wan-koo offered to resign on Monday amid a growing political scandal.

Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation. He has transferred his role of chairing Cabinet meetings to the deputy prime minister for the time being, according to his office.

Park heard about the resignation and called it "regrettable," according to the South Korean presidential office.

Calls for Lee to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul in an apparent suicide on April 9. Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials, including those who work for the President.

Lee and seven other politicians with links to the South Korean President are under investigation. cont...

South Korean PM offers resignation over bribery scandal
Suicide note leads to government bribery investigation

• Calls for Lee Wan-koo to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul
• Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials

Figure 1: A CNN news article with story highlights and side information. The second block is the main body of the article. It comes with side information such as the title (first block) and the images with their captions (third block). The last block is the story highlights that assist in gathering information on the article quickly. These highlights are often used as the gold summary of the article in summarization literature.

mentioned in the title and the image captions of the document.

In this paper, we develop a general framework for single-document summarization with side information. Our model includes a neural network-based hierarchical document encoder and a hierarchical attention-based sentence extractor. Our hierarchical document encoder resembles the architectures proposed by Cheng and Lapata (2016) and Nallapati, Zhai, and Zhou (2017), in that it derives the document meaning representation from its sentences and their constituent words. We also use recurrent neural networks to read the sequence of sentences in the document. Our novel sentence extractor combines this document meaning representation with an attention mechanism (Bahdanau, Cho, and Bengio 2014) over the side information to select sentences of the

input document as the output summary.

The idea of using additional information to improve extractive summarization is less explored. Previous work has discussed the importance of manually defined features using title words and words with pragmatic cues (e.g., "significant", "impossible" and "hardly") for summarization. Edmundson (1969) used a subjectively weighted combination of these human-engineered features, whereas Kupiec, Pedersen, and Chen (1995) and Mani (2001) trained their feature weights using a corpus. We explore the advantages of side information in a neural network-based summarization framework. Our proposed framework does not use any human-engineered features and could exploit different types of side information. In this paper, we conceptualize side information as the title of the document and the image captions present in the document.[1]

We evaluate our models automatically (in terms of ROUGE scores) on the CNN news highlights dataset (Hermann et al. 2015). Experimental results show that our summarizer informed with side information performs consistently better than the ones that do not use any side information. We also conduct a human evaluation judging which type of summary participants prefer. Our results overwhelmingly show that human subjects find our summaries more informative and complete.

2 Problem Formulation

In this section we formally define our extractive summarization problem with side information. Given a document D consisting of a sequence of sentences $(s_1, s_2, ..., s_n)$ and a sequence of pieces of side information $(c_1, c_2, ..., c_p)$, we produce a summary S by selecting m sentences from D (where $m < n$). We judge each sentence s_i for its relevance in the summary and label it with $y_i \in \{0, 1\}$ where 1 indicates that s_i should be considered for the summary and 0, otherwise. In this paper, we approach this problem in a supervised setting where we aim to maximize the likelihood of the set of labels $y = (y_1, y_2, ..., y_n)$ given the input document D and model parameters θ:

$$p(y|D; \theta) = \prod_{i=1}^{n} p(y_i|D; \theta)$$

The next section presents our model and discusses how it generates summaries informed with side information.

3 Summarization with Side Information

Our extractive summarization framework consists of a hierarchical encoder-decoder architecture assembled by recurrent neural networks (RNNs) and convolutional neural networks (CNNs). The main components of our model are a convolutional neural network sentence encoder, a recurrent

---

[1] We focus on textual side information. There are studies which show that non-textual side information could be useful as well (e.g., Hitschler, Schamoni, and Riezler 2016) in NLP. However, we leave non-textual side information for future work.
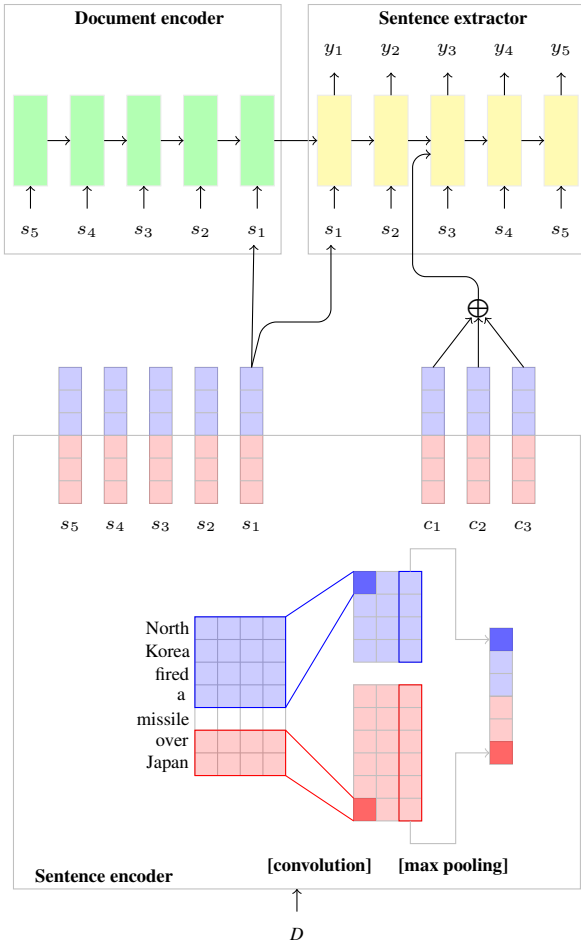
Figure 2: Hierarchical encoder-decoder model for extractive summarization with side information. $s_1, \ldots, s_5$ are sentences in the document and, c_1, c_2 and c_3 represent side information.

neural network document encoder and an attention-based recurrent neural network sentence extractor. Our model exploits the compositionality of the document. It reflects that a document is built of a meaningful sequence of sentences and each sentence is built of a meaningful sequence of words. With that in mind, we first obtain continuous representations of sentences by applying single-layer convolutional neural networks over sequences of word embeddings and then we rely on a recurrent neural network to compose sequence of sentences to get document embeddings. We model extractive summarization as a sequence labelling problem using a standard encoder-decoder architecture (Sutskever, Vinyals, and Le 2014). First, the encoder reads the sequence of sentences $(s_1, s_2, ..., s_n)$ in D and then, the decoder generates a sequence of labels $(y_1, y_2, ..., y_n)$ labelling each sentence in D. Figure 2 presents the layout of our model. In the following, we explain the main components of our model in detail.

3.1 Sentence Encoder

One core component of our hierarchical model is a convolutional sentence encoder which encodes sentences (from the main body and the side information) into continuous representations.[2] CNNs (LeCun et al. 1990) have shown to be very effective in computer vision (Krizhevsky, Sutskever, and Hinton 2012) and in NLP (Collobert et al. 2011). We chose CNNs in our framework for the following reasons. Firstly, single-layer CNNs can be trained effectively and secondly, CNNs have been shown to be effective in identifying salient patterns in the input depending on the task. For example, for the caption generation task (Xu et al. 2015), CNNs successfully identify salient objects in the image for the corresponding words in the caption. We believe that CNNs can similarly identify salient terms, e.g., named-entities and events, in sentences that correlate with the gold summary. This should in turn (i) optimize intermediate document representations in both our document encoder and sentence extractor and (ii) assist the attention mechanism to correlate salient information in the side information and sentences, for extractive summarization.

Our model is a variant of the models presented in Collobert et al. (2011), Kim (2014) and Cheng and Lapata (2016). A sentence s of length k in D is represented as a dense matrix $W = [w_1, w_2, \ldots, w_k] \in R^{k \times d}$ where $w_i \in R^d$ is the word embedding of the ith word in s. We apply a temporal narrow convolution by using a kernel filter $K \in R^{h \times d}$ of width h for a window of h words in s to produce a new feature. This filter is applied to each possible window of words in s to produce a feature map $f = [f_1, f_2, \ldots, f_{k-h+1}] \in R^{k-h+1}$ where f_i is defined:

$$f_i = \text{ReLU}(K \circ W_{i:i+h-1} + b)$$

where, $\circ$ is the Hadamard Product followed by a sum over all elements, ReLU is a rectified linear activation[3] and $b \in R$ is a bias term. We use the ReLU activation function to accelerate the convergence of stochastic gradient descent compared to sigmoid or tanh functions (Krizhevsky, Sutskever, and Hinton 2012). We then apply max pooling over time (Collobert et al. 2011) over the feature map f and get $f_{\max} = \max(f)$ as the feature corresponding to this particular filter K. Max-pooling is followed by local response normalization for better generalization (Krizhevsky, Sutskever, and Hinton 2012). We use multiple kernels K_h of width h to compute a list of features f^{K_h}. In addition, we use kernels of varying widths to learn a set of feature lists $(f^{K_{h_1}}, f^{K_{h_2}}, \ldots)$. We concatenate all feature lists to get the final sentence representation.[4]

The bottom part of Figure 2 briefly presents our convolutional sentence encoder. Kernels of sizes 2 (shown in red)

---

and 4 (shown in blue) are applied 3 times each. The max pooling over time operation leads to two feature lists f^{K_2} and $f^{K_4} \in R^3$. The final sentence embeddings have six dimensions. We use this sentence encoder to get sentence-level representations of the sentences and side information (the title and image captions) of the document D.

3.2 Document Encoder

The document encoder (shown in Figure 2, top left) composes a sequence of sentences to get a document representation. The sentence extractor, along with attending the side information, crucially exploits the document representation to identify the local and global importance of a sentence in the document to make a decision on whether it should be considered for the summary.

We use a recurrent neural network with Long Short-Term Memory (LSTM) cells to avoid the vanishing gradient problem when training long sequences (Hochreiter and Schmidhuber 1997). Given document D consisting of a sequence of sentences $(s_1, s_2, ..., s_n)$, we follow a common practice and feed sentences in reverse order (Sutskever, Vinyals, and Le 2014; Li, Luong, and Jurafsky 2015; Filippova et al. 2015). This way we make sure that the network does not omit top sentences of the document which are particularly important for summarization (Rush, Chopra, and Weston 2015; Nallapati et al. 2016). At time step t, the hidden state $h_t = \mathrm{LSTM}(s_{n-t+t}, h_{t-1})$ is updated as:

$$
\begin{bmatrix} f_t \\ i_t \\ o_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ s_{n-t+1} \end{bmatrix}
$$
$$
c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t
$$
$$
h_t = o_t \odot \tanh(c_t)
$$

where the operator $\odot$ denotes element-wise multiplication and W are the learned parameters of the model.

3.3 Sentence Extractor

Our sentence extractor (Figure 2, top right) labels each sentence in the document with labels 1 or 0 by implicitly estimating its relevance in the document and by directly attending to the side information for importance cues. It is implemented with another recurrent neural network with LSTM cells and an attention mechanism (Bahdanau, Cho, and Bengio 2014). Our attention mechanism differs from the standard practice of attending intermediate states of the input (encoder). Instead, our extractor attends to the side information in the document for cues. Given a document $D : \langle (s_1, s_2, ..., s_n), (c_1, c_2, ..., c_p) \rangle$, it reads sentences $(s_1, s_2, ..., s_n)$ in order and labels them one by one while attending the side information $(c_1, c_2, ..., c_p)$ consisting of the title and image captions. Given sentence s_t at time step t, it returns a probability distribution over labels as:

$$
p(y_t|s_t, D) = \mathrm{softmax}(g(h_t, h'_t))
$$
$$
g(h_t, h'_t) = U_o(V_h h_t + W'_h h'_t)
$$
$$
h_t = \mathrm{LSTM}(s_t, h_{t-1})
$$
$$
h'_t = \sum_{i=1}^{p} \alpha_{(t,i)} c_i,
$$
$$
\text{where } \alpha_{(t,i)} = \frac{\exp(h_t c_i)}{\sum_j \exp(h_t c_j)}
$$

where $g(\cdot)$ is a single-layer neural network with parameters U_o, V_h and W'_h. h_t is an intermediate RNN state at time step t. The dynamic context vector h'_t is essentially the weighted sum of the side information in the document. Figure 2 summarizes our model. For each labelling decision, our network considers the encoded document meaning representation, sentences labeled so far and the side information.

3.4 Summary Generation

We rank sentences in the document D by $p(1|s_t, D, \theta)$, the confidence scores assigned by the softmax layer of the sentence extractor and generate a summary S by assembling together the m best ranked sentences.

4 Experimental Setup

This section presents our experimental setup for the assessment of our models. We discuss the training and the evaluation dataset. We also explain how we augment existing datasets with side information and describe implementation details to facilitate the replication of our results. We present a brief description of our baseline systems.

4.1 Training and Test data

To train our model, we need documents annotated with sentence importance information, i.e., each sentence in a document is labelled with 1 (summary-worthy) or 0 (not summary-worthy). For our purposes, we used an augmented version of the CNN dataset (Hermann et al. 2015).[5]

Our dataset is an evolved version of the CNN dataset first collected by Svore, Vanderwende, and Burges (2007) for highlight generation. Svore, Vanderwende, and Burges (2007) noticed that CNN articles often come with "story highlights" to allow readers to quickly gather information on stories. They collected a small dataset for evaluation purposes. Woodsend and Lapata (2010) improved on this by collecting 9,000 articles and manually annotating them for sentence extraction. Recently, Hermann et al. (2015) crawled 93K CNN articles to build a large-scale corpus to

---

[5]Hermann et al. (2015) have also released the DailyMail dataset, but we do not report our results on this dataset. We found that the script written by Hermann et al. to crawl DailyMail articles mistakenly extracts image captions as part of the main body of the document. As image captions often don't have sentence boundaries, they blend with the sentences of the document unnoticeably. This leads to the production of erroneous summaries.

set a benchmark for deep learning methods. Since then, this dataset has been used for single-document summarization (Nallapati et al. 2016; Cheng and Lapata 2016; Nallapati, Zhai, and Zhou 2017; See, Liu, and Manning 2017; Tan and Wan 2017). Cheng and Lapata (2016) annotated this dataset with the Woodsend and Lapata (2010) style gold annotation using a rule-based method judging each sentence for its semantic correspondence to the gold summary. Nallapati, Zhai, and Zhou (2017) automatically extracted ground truth labels such that all positively labeled sentences from an article collectively gives the highest ROUGE score with respect to the gold summary. ROUGE (Lin and Hovy 2003), a recall-oriented metric, is often used to evaluate summarization systems. See Section 5.1 for more details. Nallapati, Zhai, and Zhou (2017) reported comparable results to Cheng and Lapata (2016) with their automatically extracted labels on the DailyMail dataset (Hermann et al. 2015).

In our experiments we annotated the CNN dataset with the Nallapati, Zhai, and Zhou (2017) style annotation. We approach this exponential problem of selecting the best subset of sentences using a greedy approach and add one sentence at a time to the summary such that the ROUGE score of the current summary is the highest with respect to the gold summary. We stop adding new sentences to the summary when the additions do not improve the ROUGE score or the maximum number of sentences in the summary is reached.[6]

We further augmented this dataset with side information. We used a modified script of Hermann et al. (2015) to extract titles and image captions, and we associated them with the corresponding articles. All articles get associated with their titles. The availability of image captions varies from 0 to 414 per article, with an average of 3 image captions. There are 40% CNN articles with at least one image caption. Our dataset is publicly available at `https://github.com/shashiongithub/sidenet`.

We trained our network on a named-entity-anonymized version of news articles.[7] However, we generate de-anonymized summaries and evaluate them against de-anonymized gold summaries to facilitate human evaluation and to make human evaluation comparable to automatic evaluation.

We used the standard splits of Hermann et al. (2015) for training, validation and testing (90K/1,220/1,093 documents respectively).

4.2   Comparison Systems

We compared the output of our model against the standard baseline of simply selecting the first three sentences from each document as the summary. We refer to this baseline as LEAD in the rest of the paper.

We also compared our system against the sentence extraction system of Cheng and Lapata (2016).[8] We refer

to this system as POINTERNET as the neural attention architecture in Cheng and Lapata (2016) resembles the one in Pointer Networks (Vinyals, Fortunato, and Jaitly 2015). It does not exploit any side information.[9] Cheng and Lapata (2016) report only on the DailyMail dataset. We used their code (`https://github.com/cheng6076/NeuralSum`) to produce results on the CNN dataset.[10]

4.3   Implementation Details

We used our training data to train word embeddings using the *Word2vec* skip-gram model (Mikolov et al. 2013) with context window size 6, negative sampling size 10 and hierarchical softmax 1. For known words, word embedding variables were initialized with pre-trained word embeddings of size 200. For unknown words, embeddings were initialized to zero, but optimized during training. All sentences, including titles and image captions, were padded with zeros to a sentence length of 100. For the convolutional sentence encoder, we followed Kim et al. (2016), and used a list of kernels of widths 1 to 7, each with output channel size of 50. This leads the sentence embedding size in our model to be 350. For the recurrent neural network component in document encoder and sentence extractor, we used a single-layered LSTM network with size 600. All input documents were padded with zeros to a maximum document length of 126. For each document, we consider a maximum of 10 image captions. We experimented with various numbers (1, 3, 5, 10 and 20) of image captions on the validation set and found that our model performed best with 10 image captions. We performed mini-batch cross-entropy training with a batch size of 20 documents for 10 training epochs. After each epoch, we evaluated our model on the validation set and chose the best performing model for the test set. We trained our models with the optimizer Adam (Kingma and Ba 2015) with initial learning rate 0.001. Our system is fully implemented in TensorFlow (Abadi et al. 2015).[11]

5   Results and Discussion

We conducted an automatic and a human evaluation. We start this section with an ablation study on the validation set. The best model from this study is chosen for the test set. In the rest of the paper, we refer to our model as SIDENET for its ability to exploit side information.

---

[6]We choose maximum three sentences in the summary. See an explanation for this in Section 5.1.

[7]We also experimented with the de-anonymized articles, but the results were inferior to those presented here with the anonymized data.

[8]The architecture of POINTERNET is closely related to the ar-

chitecture of SIDENET without side information.

[9]Adding side information to POINTERNET is an interesting direction of research but we do not pursue it here. It requires decoding with multiple types of attentions, and this is not the focus of this paper.

[10]We are unable to compare our results to the extractive system of Nallapati, Zhai, and Zhou (2017) because they report their results on the DailyMail dataset and their code is not available. The abstractive systems of Chen et al. (2016) and Tan and Wan (2017) report their results on the CNN dataset, however, their results are not comparable to ours as they report on the full-length F_1 variants of ROUGE to evaluate their abstractive summaries. We report ROUGE recall scores which is more appropriate to evaluate our extractive summaries.

[11]Our TensorFlow code is publicly available at `https://github.com/shashiongithub/sidenet`.

MODELS	R1	R2	R3	R4	RL	Avg.
LEAD	49.2	18.9	9.8	6.0	43.8	25.5
POINTERNET	53.3	19.7	10.4	6.4	47.2	27.4
SIDENET+TITLE	55.0	21.6	11.7	7.5	48.9	28.9
SIDENET+CAPTION	55.3	21.3	11.4	7.2	49.0	28.8
SIDENET+FS	54.8	21.1	11.3	7.2	48.6	28.6
Combination Models (SIDENET+)						
TITLE+CAPTION	**55.4**	**21.8**	**11.8**	**7.5**	**49.2**	**29.2**
TITLE+FS	55.1	21.6	11.6	7.4	48.9	28.9
CAPTION+FS	55.3	21.5	11.5	7.3	49.0	28.9
TITLE+CAPTION+FS	55.4	21.5	11.6	7.4	49.1	29.0

Table 1: Ablation results on the validation set. We report R1, R2, R3, R4, RL and their average (Avg.). The first block of the table presents LEAD and POINTERNET which do not use any side information. LEAD is the baseline system selecting "first" three sentences. POINTERNET is the sentence extraction system of Cheng and Lapata. SIDENET is our model. The second and third blocks of the table present different variants of SIDENET. We experimented with three types of side information: title (TITLE), image captions (CAPTION) and the first sentence (FS) of the document. The bottom block of the table presents models with more than one type of side information. The best performing model (highlighted in boldface) is used on the test set.

MODELS	R1	R2	R3	R4	RL
Fixed length: 75b					
LEAD	20.1	7.1	**3.5**	2.1	14.6
POINTERNET	**20.3**	**7.2**	**3.5**	**2.2**	**14.8**
SIDENET	20.2	7.1	3.4	2.0	14.6
Fixed length: 275b					
LEAD	39.1	14.5	7.6	4.7	34.6
POINTERNET	38.6	13.9	7.3	4.4	34.3
SIDENET	**39.7**	**14.7**	**7.9**	**5.0**	**35.2**
Full length summaries					
LEAD	49.3	19.5	10.7	6.9	43.8
POINTERNET	51.7	19.7	10.6	6.6	45.7
SIDENET	**54.2**	**21.6**	**12.0**	**7.9**	**48.1**

Table 2: Final results on the test set. POINTERNET is the sentence extraction system of Cheng and Lapata. SIDENET is our best model from Table 1. Best ROUGE score in each block and each column is highlighted in boldface.

5.1 Automatic Evaluation

To automatically assess the quality of our summaries, we used ROUGE (Lin and Hovy 2003), a recall-oriented metric, to compare our model-generated summaries to manually-written highlights.[12] Previous work has reported ROUGE-1 (R1) and ROUGE-2 (R2) scores to access informativeness, and ROUGE-L (RL) to access fluency. In addition to R1, R2 and RL, we also report ROUGE-3 (R3) and ROUGE-4 (R4) capturing higher order n-grams overlap to assess informativeness and fluency simultaneously.

We follow Cheng and Lapata and report on both full length (three sentences with the top scores as the summary)

---

[12] We used pyrouge, a Python package, to compute all our rouge scores with parameters "-a -c 95 -m -n 4 -w 1.2."

and fixed length (first 75 bytes and 275 bytes as the summary) summaries. For full length summaries, our decision of selecting three sentences is guided by the fact that there are 3.11 sentences on average in the gold highlights of the training set. We conduct our ablation study on the validation set with full length ROUGE scores, but we report both fixed and full length ROUGE scores for the test set.

We experimented with two types of side information: title (TITLE) and image captions (CAPTION). In addition, we experimented with the first sentence (FS) of the document as side information. Note that the latter is not strictly speaking side information, it is a sentence in the document. However, we wanted to explore the idea that the first sentence of the document plays a crucial part in generating summaries (Rush, Chopra, and Weston 2015; Nallapati et al. 2016). SIDENET with FS acts as a baseline for SIDENET with title and image captions.

We report the performance levels of several variants of SIDENET on the validation set in Table 1. We also compare them against the LEAD baseline and POINTERNET. These two systems do not use any side information. Interestingly, all the variants of SIDENET significantly outperform LEAD and POINTERNET. When the title (TITLE), image captions (CAPTION) and the first sentence (FS) are used separately as side information, SIDENET performs best with TITLE as its side information. Our result demonstrates the importance of the title of the document in extractive summarization (Edmundson 1969; Kupiec, Pedersen, and Chen 1995; Mani 2001). The performance with TITLE and CAPTION is better than that with FS. We also tried possible combinations of TITLE, CAPTION and FS. All SIDENET models are superior to the ones without any side information. SIDENET performs best when TITLE and CAPTION are jointly used as side information (55.4%, 21.8%, 11.8%, 7.5%, and 49.2% for R1, R2, R3, R4, and RL respectively). It is better than the the LEAD baseline by 3.7 points on average and than POINTERNET by 1.8 points on average, indicating that side information is useful to identify the gist of the document. We use this model for testing purposes.

Our final results on the test set are shown in Table 2. We present both fixed length (first 75 bytes and 275 bytes) and full length (three highest scoring sentences) ROUGE scores. It turns out that for smaller summaries (75 bytes) LEAD and POINTERNET are superior to SIDENET. This result could be because LEAD (always) and POINTERNET (often) include the first sentence in their summaries, whereas, SIDENET is better capable at selecting sentences from various document positions. This is not captured by smaller summaries of 75 bytes, but it becomes more evident with longer summaries (275 bytes and full length) where SIDENET performs best across all ROUGE scores. It is interesting to note that POINTERNET performs better than LEAD for 75-byte summaries, then its performance drops behind LEAD for 275-byte summaries, but then it performs better than LEAD for full length summaries for R1, R2 and RL. It shows that POINTERNET with its attention over sentences in the document is capable of exploring more than first few sentences in the document. But, it is still behind SIDENET which is better at identifying salient sentences in the document. SIDENET

Models	1st	2nd	3rd	4th
LEAD	0.15	0.17	**0.47**	0.21
POINTERNET	0.16	0.05	0.31	**0.48**
SIDENET	0.28	**0.53**	0.15	0.04
HUMAN	**0.41**	0.25	0.07	0.27

Table 3: Human evaluations: Ranking of various systems. Rank 1st is best and rank 4th, worst. Numbers show the percentage of times a system gets ranked at a certain position.

LEAD	• Seoul South korea's Prime Minister Lee Wan-koo offered to resign on monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • He has transferred his role of chairing cabinet meetings to the deputy Prime Minister for the time being , according to his office
POINTERNET	• South Korea's Prime Minister Lee Wan-koo offered to resign on Monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • Lee and seven other politicians with links to the South Korean President are under investigation
SIDENET	• South Korea's Prime Minister Lee Wan-Koo offered to resign on Monday amid a growing political scandal • Lee will stay in his official role until South Korean President Park Geun-hye accepts his resignation • Calls for Lee to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul in an apparent suicide on April 9
HUMAN	• Calls for Lee Wan-koo to resign began after South Korean tycoon Sung Woan-jong was found hanging from a tree in Seoul • Sung, who was under investigation for fraud and bribery, left a note listing names and amounts of cash given to top officials

Figure 3: Summaries produced by various systems for the article shown in Figure 1.

performs better than POINTERNET by 0.8 points for 275-byte summaries and by 1.9 points for full length summaries, on average for all ROUGE scores.

5.2 Human Evaluation

We complement our automatic evaluation results with human evaluation. We randomly selected 20 articles from the test set. Annotators were presented with a news article and summaries from four different systems. These include the LEAD baseline, POINTERNET, SIDENET and the human authored highlights. We followed the guidelines in Cheng and Lapata (2016), and asked our participants to rank the summaries from best (1st) to worst (4th) in order of informativeness (does the summary capture important information in the article?) and fluency (is the summary written in well-formed English?). We did not allow any ties, we only sampled articles with non-identical summaries. We assigned this task to five annotators who were proficient English speakers. Each annotator was presented with all 20 articles. The order of summaries to rank was randomized per article. Examples of summaries our subjects ranked are shown in Figure 3.

The results of our human evaluation study are shown in Table 3. We compare our SIDENET against LEAD, POINTERNET and HUMAN on how frequently each system gets ranked 1st, 2nd and so on, in terms of best-to-worst summaries. As one might imagine, HUMAN gets ranked 1st most of the time (41%). However, it is closely followed by SIDENET with ranked 1st 28% of the time. In comparison, POINTERNET and LEAD were mostly ranked at 3rd and 4th places. We also carried out pairwise comparisons between all models in Table 3 for their statistical significance using a one-way ANOVA with post-hoc Tukey HSD tests with ($p < 0.01$). It showed that SIDENET is significantly better than LEAD and POINTERNET, and it does not differ significantly from HUMAN. On the other hand, POINTERNET does not differ significantly from LEAD and it differs significantly from both SIDENET and HUMAN. The human evaluation results corroborates our empirical results in Table 1 and Table 2: SIDENET is better than LEAD and POINTERNET in producing informative and fluent summaries.

Figure 3 shows output summaries from various systems for the article shown in Figure 1. As can be seen, both SIDENET and POINTERNET were able to select the most relevant sentence for the summary from anywhere in the article, but SIDENET is better at producing summaries which are close to human authored summaries.

6 Conclusion

In this paper, we developed a neural network framework for single-document extractive summarization with side information. We evaluated our system on the large scale CNN dataset. Our experiments show that side information is useful for extracting salient sentences from the document for the summary. Our framework is very general and it could exploit different types of side information. There are few previous works which improve extractive summarization with external knowledge from third party sources. Svore, Vanderwende, and Burges (2007) included features from news search query logs and Wikipedia entities to summarize CNN articles. Recently, Li et al. (2016) used public posts following a news article to improve automatic summarization. For future work, it would be interesting to use such knowledge as side information in our framework.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR-2015 (abs/1409.0473)*.

Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; and Jiang, H. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of IJCAI*, 2754–2760.

Cheng, J., and Lapata, M. 2016. Neural summarization by extracting sentences and words. In *Proceedings of ACL*, 484–494.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

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

Filatova, E., and Hatzivassiloglou, V. 2004. Event-based extractive summarization. In *Proceedings of ACL Workshop on Text Summarization Branches Out*, 104–111.

Filippova, K.; Alfonseca, E.; Colmenares, C. A.; Kaiser, L.; and Vinyals, O. 2015. Sentence compression by deletion with LSTMs. In *Proceedings of EMNLP*, 360–368.

Hermann, K. M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. In *NIPS 28*, 1693–1701.

Hitschler, J.; Schamoni, S.; and Riezler, S. 2016. Multimodal pivots for image caption translation. In *Proceedings of ACL*, 2399–2409.

Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9(8):1735–1780.

Kim, Y.; Jernite, Y.; Sontag, D.; and Rush, A. M. 2016. Character-aware neural language models. In *Proceedings of AAAI*, 2741–2749.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, 1746–1751.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Kågebäck, M.; Mogren, O.; Tahmasebi, N.; and Dubhashi, D. 2014. Extractive summarization using continuous vector space models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, 31–39.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS 25*, 1097–1105.

Kupiec, J.; Pedersen, J.; and Chen, F. 1995. A trainable document summarizer. In *Proceedings of SIGIR*, 406–407.

Le, Q. V., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*, 1188–1196.

LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Habbard, W. E.; and Jackel, L. D. 1990. Handwritten digit recognition with a back-propagation network. In *NIPS 2*, 396–404.

Li, C.; Wei, Z.; Liu, Y.; Jin, Y.; and Huang, F. 2016. Using relevant public posts to enhance news article summarization. In *Proceedings of COLING*, 557–566.

Li, J.; Luong, T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of ACL*, 1106–1115.

Liepins, R.; Germann, U.; Barzdins, G.; Birch, A.; Renals, S.; Weber, S.; van der Kreeft, P.; Bourlard, H.; Prieto, J. a.; Klejch, O.; Bell, P.; Lazaridis, A.; Mendes, A.; Riedel, S.; Almeida, M. S. C.; Balage, P.; Cohen, S. B.; Dwojak, T.; Garner, P. N.; Giefer, A.; Junczys-Dowmunt, M.; Imran, H.; Nogueira, D.; Ali, A.; Miranda, S. a.; Popescu-Belis, A.; Miculicich Werlen, L.; Papasarantopoulos, N.; Obamuyide, A.; Jones, C.; Dalvi, F.; Vlachos, A.; Wang, Y.; Tong, S.; Sennrich, R.; Pappas, N.; Narayan, S.; Damonte, M.; Durrani, N.; Khurana, S.; Abdelali, A.; Sajjad, H.; Vogel, S.; Sheppey, D.; Hernon, C.; and Mitchell, J. 2017. The summa platform prototype. In *Proceedings of EACL: Software Demonstrations*, 116–119.

Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NAACL*, 71–78.

Mani, I. 2001. *Automatic Summarization*. Natural language processing. John Benjamins Publishing Company.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS 26*, 3111–3119.

Nallapati, R.; Zhou, B.; dos Santos, C. N.; Gülçehre, Ç.; and Xiang, B. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of CoNLL*, 280–290.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of AAAI*, 3075–3081.

Nenkova, A.; Vanderwende, L.; and McKeown, K. 2006. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of ACM SIGIR*, 573–580.

Radev, D.; Allison, T.; Blair-Goldensohn, S.; Blitzer, J.; Çelebi, A.; Dimitrov, S.; Drabek, E.; Hakim, A.; Lam, W.; Liu, D.; Otterbacher, J.; Qi, H.; Saggion, H.; Teufel, S.; Topper, M.; Winkel, A.; and Zhang, Z. 2004. MEAD — A plat-

form for multidocument multilingual text summarization. In *Proceedings of LREC*, 699–702.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP*, 379–389.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*, 1073–1083.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS 27*, 3104–3112.

Svore, K. M.; Vanderwende, L.; and Burges, C. J. C. 2007. Enhancing single-document summarization by combining ranknet and third-party sources. In *Proceedings of EMNLP-CoNLL*, 448–457.

Tan, J., and Wan, X. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of ACL*, 1171–1181.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *NIPS 28*, 2692–2700.

Woodsend, K., and Lapata, M. 2010. Automatic generation of story highlights. In *Proceedings of ACL*, 565–574.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*, 2048–2057.

Yasunaga, M.; Zhang, R.; Meelu, K.; Pareek, A.; Srinivasan, K.; and Radev, D. 2017. Graph-based neural multi-document summarization. In *Proceedings of CoNLL*, 452–462.

Yin, W., and Pei, Y. 2015. Optimizing sentence modeling and selection for document summarization. In *Proceedings of IJCAI*, 1383–1389.