AttSum: Joint Learning of Focusing and Summarization with Neural Attention

Ziqiang Cao[1] **Wenjie Li**[1] **Sujian Li**[2] **Furu Wei**[3] **Yanran Li**[1]
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]Key Laboratory of Computational Linguistics, Peking University, MOE, China
[3]Microsoft Research, Beijing, China
{cszqcao, cswjli, csyli}@comp.polyu.edu.hk
lisujian@pku.edu.cn
furu@microsoft.com

Abstract

Query relevance ranking and sentence saliency ranking are the two main tasks in extractive query-focused summarization. Previous supervised summarization systems often perform the two tasks in isolation. However, since reference summaries are the trade-off between relevance and saliency, using them as supervision, neither of the two rankers could be trained well. This paper proposes a novel summarization system called AttSum, which tackles the two tasks jointly. It automatically learns distributed representations for sentences as well as the document cluster. Meanwhile, it applies the attention mechanism to simulate the attentive reading of human behavior when a query is given. Extensive experiments are conducted on DUC query-focused summarization benchmark datasets. Without using any hand-crafted features, AttSum achieves competitive performance. We also observe that the sentences recognized to focus on the query indeed meet the query need.

1 Introduction

Query-focused summarization (Dang, 2005) aims to create a brief, well-organized and fluent summary that answers the need of the query. It is useful in many scenarios like news services and search engines, etc. Nowadays, most summarization systems are under the extractive framework which directly selects existing sentences to form the summary. Basically, there are two major tasks in extractive query-focused summarization, i.e., to measure the saliency of a sentence and its relevance to a user's query.

After a long period of research, learning-based models like Logistic Regression (Li et al., 2013) etc. have become growingly popular in this area. However, most current supervised summarization systems often perform the two tasks in isolation. Usually, they design query-dependent features (e.g., query word overlap) to learn the relevance ranking, and query-independent features (e.g., term frequency) to learn the saliency ranking. Then, the two types of features are combined to train an overall ranking model. Note that the only supervision available is the reference summaries. Humans write summaries with the trade-off between relevance and saliency. Some salient content may not appear in reference summaries if it fails to respond to the query. Likewise, the content relevant to the query but not representative of documents will be excluded either. As a result, in an isolated model, weights for neither query-dependent nor query-independent features could be learned well from reference summaries.

In addition, when measuring the query relevance, most summarization systems merely make use of surface features like the TF-IDF cosine similarity between a sentence and the query (Wan and Xiao, 2009). However, relevance is not similarity. Take the document cluster "d360f" in DUC[1] 2005 as an example. It has the following query: *What are the benefits of drug legalization?* Here, "Drug legalization" are the key words with high TF-IDF scores. And yet the main intent of the query is to look for "benefit", which is a very general word and does not present in the source text at all. It is not surprising that when measured by the TF-IDF cosine similarity, the sentences with top scores all contain the words "drug" or "legalization". Nevertheless, none of them provides advantages of drug legalization. See Section 4.6

[1] http://www-nlpir.nist.gov/projects/duc/

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 547–556, Osaka, Japan, December 11-17 2016.

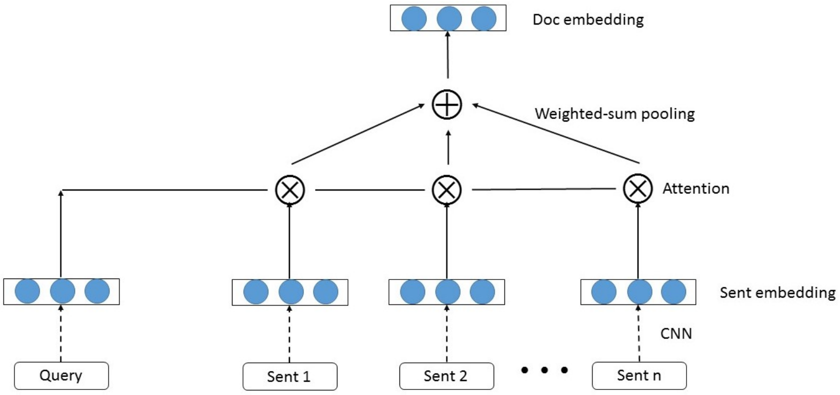

Figure 1: Generation of sentence and document cluster embeddings. "$\oplus$" stands for a pooling operation, while "$\otimes$" represents a relevance measurement function.

for reference. Apparently, even if a sentence is exactly the same as the query, it is still totally useless in the summary because it is unable to answer the query need. Therefore, the surface features are inadequate to measure the query relevance, which further augments the error of the whole summarization system. This drawback partially explains why it might achieve acceptable performance to adopt generic summarization models in the query-focused summarization task (e.g., (Gillick and Favre, 2009)).

Intuitively, the isolation problem can be solved with a joint model. Meanwhile, neural networks have shown to generate better representations than surface features in the summarization task (Cao et al., 2015b; Yin and Pei, 2015). Thus, a joint neural network model should be a nice solution to extractive query-focused summarization. To this end, we propose a novel summarization system called AttSum, which joints query relevance ranking and sentence saliency ranking with a neural attention model. The attention mechanism has been successfully applied to learn alignment between various modalities (Chorowski et al., 2014; Xu et al., 2015; Bahdanau et al., 2014). In addition, the work of (Kobayashi et al., 2015) demonstrates that it is reasonably good to use the similarity between the sentence embedding and document embedding for saliency measurement, where the document embedding is derived from the sum pooling of sentence embeddings. In order to consider the relevance and saliency simultaneously, we introduce the weighted-sum pooling over sentence embeddings to represent the document, where the weight is the automatically learned query relevance of a sentence. In this way, the document representation will be biased to the sentence embeddings which match the meaning of both query and documents. The working mechanism of AttSum is consistent with the way how humans read when having a particular query in their minds. Naturally, they pay more attention to the sentences that meet the query need. It is noted that, unlike most previous summarization systems, our model is totally data-driven, i.e., all the features are learned automatically.

We verify AttSum on the widely-used DUC 2005 $\sim$ 2007 query-focused summarization benchmark datasets. AttSum outperforms widely-used summarization systems which rely on rich hand-crafted features. We also conduct qualitative analysis for those sentences with large relevance scores to the query. The result reveals that AttSum indeed focuses on highly query relevant content.

The contributions of our work are as follows:

- We apply the attention mechanism that tries to simulate human attentive reading behavior for query-focused summarization;

- We propose a joint neural network model to learn query relevance ranking and sentence saliency ranking simultaneously.

2 Query-Focused Sentence Ranking

For generic summarization, people read the text with almost equal attention. However, given a query, people will naturally pay more attention to the query relevant sentences and summarize the main ideas from them. Similar to human attentive reading behavior, AttSum, the system to be illustrated in this section, ranks the sentences with its focus on the query. The overall framework is shown in Fig. 1. From the bottom to up, AttSum is composed of three major layers.

CNN Layer Use Convolutional Neural Networks to project the sentences and queries onto the embeddings.

Pooling Layer With the attention mechanism, combine the sentence embeddings to form the document embedding in the same latent space.

Ranking Layer Rank a sentence according to the similarity between its embedding and the embedding of the document cluster.

The rest of this section describes the details of the three layers.

2.1 CNN Layer

Convolutional Neural Networks (CNNs) have been widely used in various Natural Language Processing (NLP) areas including summarization (Cao et al., 2015b; Yin and Pei, 2015). They are able to learn the compressed representations of n-grams effectively and tackle the sentences with variable lengths naturally. We use CNNs to project both sentences and the query onto distributed representations, i.e.,

$$\mathbf{v}(s) = \text{CNN}(s)$$
$$\mathbf{v}(q) = \text{CNN}(q)$$

A basic CNN contains a convolution operation on the top of word embeddings, which is followed by a pooling operation. Let $\mathbf{v}(w_i) \in \mathbb{R}^k$ refer to the k-dimensional word embedding corresponding to the i_{th} word in the sentence. Assume $\mathbf{v}(w_i : w_{i+j})$ to be the concatenation of word embeddings $[\mathbf{v}(w_i), \cdots, \mathbf{v}(w_{i+j})]$. A convolution operation involves a filter $\mathbf{W}_t^h \in \mathbb{R}^{l \times hk}$, which is applied to a window of h words to produce the abstract features $\mathbf{c}_i^h \in \mathbb{R}^l$:

$$\mathbf{c}_i^h = f(\mathbf{W}_t^h \times \mathbf{v}(w_i : w_{i+j})), \tag{1}$$

where $f(\cdot)$ is a non-linear function and the use of $tanh$ is the common practice. To simplify, the bias term is left out. This filter is applied to each possible window of words in the sentence to produce a feature map. Subsequently, a pooling operation is applied over the feature map to obtain the final features $\hat{\mathbf{c}}^h \in \mathbb{R}^l$ of the filter. Here we use the max-over-time pooling (Collobert et al., 2011).

$$\hat{\mathbf{c}}^h = \max\{\mathbf{c}_1^h, \mathbf{c}_2^h, \cdots\} \tag{2}$$

The idea behind it is to capture the most important features in a feature map. $\hat{\mathbf{c}}^h$ is the output of CNN Layer, i.e., the embeddings of sentences and queries.

2.2 Pooling Layer

With the attention mechanism, AttSum uses the weighted-sum pooling over the sentence embeddings to represent the document cluster. To achieve this aim, AttSum firstly learns the query relevance of a sentence automatically:

$$r(s, q) = \sigma(\mathbf{v}(s)\mathbf{M}\mathbf{v}(q)^T), \tag{3}$$

where $\mathbf{v}(s)\mathbf{M}\mathbf{v}(q)^T, \mathbf{M} \in \mathbb{R}^{l \times l}$ is a tensor function, and σ stands for the sigmoid function. The tensor function has the power to measure the interaction between any two elements of sentence and query embeddings. Therefore, two identical embeddings will have a low score. This characteristic is exactly what we need. To reiterate, relevance is not equivalent to similarity. Then with $r(s, q)$ as weights, we introduce the weighted-sum pooling to calculate the document embedding $\mathbf{v}(d|q)$:

$$\mathbf{v}(d|q) = \sum\nolimits_{s \in d} r(s, q)\mathbf{v}(s) \tag{4}$$

Notably, a sentence embedding plays two roles, both the pooling item and the pooling weight. On the one hand, if a sentence is highly related to the query, its pooling weight is large. On the other hand, if a sentence is salient in the document cluster, its embedding should be representative. As a result, the weighted-sum pooling generates the document representation which is automatically biased to embeddings of sentences match both documents and the query.

AttSum simulates human attentive reading behavior, and the attention mechanism in it has actual meaning. The experiments to be presented in Section 4.6 will demonstrate its strong ability to catch query relevant sentences. Actually, the attention mechanism has been applied in one-sentence summary generation before (Rush et al., 2015; Hu et al., 2015). The success of these works, however, heavily depends on the hand-crafted features. We believe that the attention mechanism may not be able to play its anticipated role if it is not used appropriately.

2.3 Ranking Layer

Since the semantics directly lies in sentence and document embeddings, we rank a sentence according to its embedding similarity to the document cluster, following the work of (Kobayashi et al., 2015). Here we adopt cosine similarity:

$$\cos(d, s|q) = \frac{\mathbf{v}(s) \bullet \mathbf{v}(d|q)^T}{||\mathbf{v}(s)|| \bullet ||\mathbf{v}(d|q)||} \tag{5}$$

Compared with Euclidean distance, one advantage of cosine similarity is that it is automatically scaled. According to (Kågebäck et al., 2014), cosine similarity is the best metrics to measure the embedding similarity for summarization.

In the training process, we apply the pairwise ranking strategy (Collobert et al., 2011) to tune model parameters. Specifically, we calculate the ROUGE-2 scores (Lin, 2004) of all the sentences in the training dataset. Those sentences with high ROUGE-2 scores are regarded as positive samples, and the rest as negative samples. Afterwards, we randomly choose a pair of positive and negative sentences which are denoted as s^+ and s^-, respectively. Through the CNN Layer and Pooling Layer, we generate the embeddings of $\mathbf{v}(s^+)$, $\mathbf{v}(s^-)$ and $\mathbf{v}(d|q)$. We can then obtain the ranking scores of s^+ and s^- according to Eq. 5. With the pairwise ranking criterion, AttSum should give a positive sample a higher score in comparison with a negative sample. The cost function is defined as follows:

$$\begin{aligned} &\epsilon(d, s^+, s^-|q) \\ &= \max(0, \Omega - \cos(d, s^+|q) + \cos(d, s^-|q)), \end{aligned} \tag{6}$$

where Ω is a margin threshold. With this cost function, we can use the gradient descent algorithm to update model parameters. In this paper, we apply the diagonal variant of AdaGrad with mini-batches (Duchi et al., 2011). AdaGrad adapts the learning rate for different parameters at different steps. Thus it is less sensitive to initial parameters than the stochastic gradient descent.

3 Sentence Selection

A summary is obliged to offer both informative and non-redundant content. While AttSum focuses on sentence ranking, it employs a simple greedy algorithm, similar to the MMR strategy (Carbonell and Goldstein, 1998), to select summary sentences. At first, we discard sentences less than 8 words like the work of (Erkan and Radev, 2004). Then we sort the rest in descending order according to the derived ranking scores. Finally, we iteratively dequeue the top-ranked sentence, and append it to the current summary if it is non-redundant. A sentence is considered non-redundant if it contains significantly new bi-grams compared with the current summary content. We empirically set the cut-off of the new bi-gram ratio to 0.5.

4 Experiments

4.1 Dataset

In this work, we focus on the query-focused multi-document summarization task. The experiments are conducted on the DUC 2005 $\sim$ 2007 datasets. All the documents are from news websites and grouped into various thematic clusters. In each cluster, there are four reference summaries created by NIST assessors. We use Stanford CoreNLP[2] to process the datasets, including sentence splitting and tokenization. Our summarization model compiles the documents in a cluster into a single document. Table 1 shows the basic information of the three datasets. We can find that the data sizes of DUC are quite different. The sentence number of DUC 2007 is only about a half of DUC 2005's. For each cluster, a summarization system is requested to generate a summary with the length limit of 250 words. We conduct a 3-fold cross-validation on DUC datasets, with two years of data as the training set and one year of data as the test set.

Year	Clusters	Sentences	Data Source
2005	50	45931	TREC
2006	59	34560	AQUAINT
2007	30	24282	AQUAINT

Table 1: Statistics of the DUC datasets.

4.2 Model Setting

For the CNN layer, we introduce a word embedding set which is trained on a large English news corpus (10^{10} tokens) with the word2vec model (Mikolov et al., 2013). The dimension of word embeddings is set to 50, like many previous work (e.g., (Collobert et al., 2011)). Since the summarization dataset is quite limited, we do not update these word embeddings in the training process, which greatly reduces the model parameters to be learned. There are two hyper-parameters in our model, i.e., the word window size h and the CNN layer dimension l. We set $h = 2$, which is consistent with the ROUGE-2 evaluation. As for l, we explore the change of model performance with $l \in [5, 100]$. Finally, we choose $l = 50$ for all the rest experiments. It is the same dimension as the word embeddings. During the training of pairwise ranking, we set the margin $\Omega = 0.5$. The initial learning rate is 0.1 and batch size is 100.

4.3 Evaluation Metric

For evaluation, we adopt the widely-used automatic evaluation metric ROUGE (Lin, 2004) [3]. It measures the summary quality by counting the overlapping units such as the n-grams, word sequences and word pairs between the peer summary and reference summaries. We take ROUGE-2 as the main measures due to its high capability of evaluating automatic summarization systems (Owczarzak et al., 2012). During the training data of pairwise ranking, we also rank the sentences according to ROUGE-2 scores.

4.4 Baselines

To evaluate the summarization performance of AttSum, we implement rich extractive summarization methods. Above all, we introduce two common baselines. The first one just selects the leading sentences to form a summary. It is often used as an official baseline of DUC, and we name it "LEAD". The other system is called "QUERY_SIM", which directly ranks sentences according to its TF-IDF cosine similarity to the query. In addition, we implement two popular extractive query-focused summarization methods, called MultiMR (Wan and Xiao, 2009) and SVR (Ouyang et al., 2011). MultiMR is a graph-based manifold ranking method which makes uniform use of the sentence-to-sentence relationships and the sentence-to-query relationships. SVR extracts both query-dependent and query-independent features and applies Support Vector Regression to learn feature weights. Note that MultiMR is unsupervised while SVR is supervised. Since our model is totally data-driven, we introduce a recent summarization system DocEmb (Kobayashi et al., 2015) that also just use deep neural network features to rank sentences. It initially works for generic summarization and we supplement the query information to compute the document representation.

[2] http://stanfordnlp.github.io/CoreNLP/

[3] ROUGE-1.5.5 with options: -n 2 -m -u -c 95 -l 250 -x -r 1000 -f A -p 0.5 -t 0

To verify the effectiveness of the joint model, we design a baseline called ISOLATION, which performs saliency ranking and relevance ranking in isolation. Specifically, it directly uses the sum pooling over sentence embeddings to represent the document cluster. Therefore, the embedding similarity between a sentence and the document cluster could only measure the sentence saliency. To include the query information, we supplement the common hand-crafted feature TF-IDF cosine similarity to the query. This query-dependent feature, together with the embedding similarity, are used in sentence ranking. ISOLATION removes the attention mechanism, and mixtures hand-crafted and automatically learned features. All these methods adopt the same sentence selection process illustrated in Section 3 for a fair comparison.

4.5 Summarization Performance

The ROUGE scores of the different summarization methods are presented in Table 2. We consider ROUGE-2 as the main evaluation metrics, and also provide the ROUGE-1 results as the common practice. As can be seen, AttSum always enjoys a reasonable increase over ISOLATION, indicating that the joint model indeed takes effects. With respect to other methods, AttSum largely outperforms two baselines (LEAD and QUERY_SIM) and the unsupervised neural network model DocEmb. Although AttSum is totally data-driven, its performance is better than the widely-used summarization systems MultiMR and SVR. It is noted that SVR heavily depends on hand-crafted features. Nevertheless, AttSum almost outperforms SVR all the time. The only exception is DUC 2005 where AttSum is slightly inferior to SVR in terms of ROUGE-2. Over-fitting is a possible reason. Table 1 demonstrates the data size of DUC 2005 is highly larger than the other two. As a result, when using the 3-fold cross-validation, the number of training data for DUC 2005 is the smallest among the three years. The lack of training data impedes the learning of sentence and document embeddings.

It is interesting that ISOLATION achieves competitive performance but DocEmb works terribly. The pre-trained word embeddings seem not to be able to measure the sentence saliency directly. In comparison, our model can learn the sentence saliency well.

Year	Model	ROUGE-1	ROUGE-2
2005	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
2006	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**
2007	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**

Table 2: ROUGE scores (%) of different models. We draw a line to distinguish models with or without hand-crafted features.

4.6 Query Relevance Performance

We check the feature weights in SVR and find the query-dependent features hold extremely small weights. Without these features, the performance of SVR only drops 1%. Therefore, SVR fails to learn query relevance well. The comparison of AttSum and ISOLATION has shown that our method

AttSum	It acknowledges that illegal drugs cannot be kept out of the country by tougher border control and interdiction measures.
	Much greater resources, derived from taxation of the drugs that are now illegal and untaxed and from the billions saved by not wasting money on more criminal- justice measures, must be devoted to drug treatment and drug prevention.
	As is the case with tobacco, legalizing marijuana, cocaine and heroin would not signify an endorsement of their use.
	The consumption and production of marijuana in the United States is on the decrease, and that criminalization costs society more in terms of increased law-enforcement-related costs and deprived revenues from taxes on pot than legalization would.
TF-IDF	Drug prices have soared.
	Drug addicts are not welcome.
	How refreshing to have so much discourse on drugs and legalization.
	The only solution now is a controlled policy of drug legalization.
Query	What are the benefits of drug legalization?
AttSum	Boparai also said that wetlands in many developing countries were vital to the sustenance of human beings, not just flora and fauna.
	EPA says that all water conservation projects, and agriculture and forestry development along China's major rivers must be assessed in accordance with environmental protection standards, and that no projects will be allowed if they pose a threat to the environment.
	Finland has agreed to help central China's Hunan Province improve biodiversity protection, environmental education, subtropical forestry and wetlands protection, according to provincial officials.
	The EPA had sought as early 1993 to subject all development on wetlands to strict environmental review, but that approach was rejected by the courts, which ruled in favor of arguments made by developers and by the National Mining Association.
TF-IDF	Statistics on wetlands loss vary widely.
	Mitigation of any impact on wetlands by creating or enhancing other wetlands.
	The new regulations would cover about one-fourth of all wetlands.
	Now more and more people have recognized wetlands' great ecological and economic potential and the conservation and utilization of wetlands has become an urgent task.
Query	Why are wetlands important? Where are they threatened? What steps are being taken to preserve them? What frustrations and setbacks have there been?

Table 3: Sentences recognized to focus on the query.

can learn better query relevance than hand-crafted features. In this section, we perform the qualitative analysis to inspect what AttSum actually catches according to the learned query relevance. We randomly choose some queries in the test datasets and calculate the relevance scores of sentences according to Eq. 3. We then extract the top ranked sentences and check whether they are able to meet the query need. Examples for both one-sentence queries and multiple-sentence queries are shown in Table 3. We also give the sentences with top TF-IDF cosine similarity to the query for comparison.

With manual inspection, we find that most query-focused sentences in AttSum can answer the query to a large extent. For instance, when asked to tell the advantages of drug legalization, AttSum catches the sentences about drug trafficking prevention, the control of marijuana use, and the economic effectiveness, etc. All these aspects are mentioned in reference summaries. The sentences with the high TF-IDF similarity, however, are usually short and simply repeat the key words in the query. The advantage of AttSum over TF-IDF similarity is apparent in query relevance ranking.

When there are multiple sentences in a query, AttSum may only focus on a part of them. Take the second query in Table 3 as an example. Although the responses to all the four query sentences are involved more or less, we can see that AttSum tends to describe the steps of wetland preservation more. Actually, by inspection, the reference summaries do not treat the query sentences equally either. For this query, they only tell a little about frustrations during wetland preservation. Since AttSum projects a query onto a single embedding, it may augment the bias in reference summaries. It seems to be hard even for humans to read attentively when there are a number of needs in a query. Because only a small part of DUC datasets contains such a kind of complex queries, we do not purposely design a special model to handle them in our current work.

5 Related Work

5.1 Extractive Summarization

Work on extractive summarization spans a large range of approaches. Starting with unsupervised methods, one of the widely known approaches is Maximum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). It used a greedy approach to select sentences and considered the trade-off between saliency and redundancy. Good results could be achieved by reformulating this as an Integer Linear Programming (ILP) problem which was able to find the optimal solution (McDonald, 2007; Gillick and Favre, 2009). Graph-based models played a leading role in the extractive summarization area, due to its ability to reflect various sentence relationships. For example, (Wan and Xiao, 2009) adopted manifold ranking to make use of the within-document sentence relationships, the cross-document sentence relationships and the sentence-to-query relationships. In contrast to these unsupervised approaches, there are also various learning-based summarization systems. Different classifiers have been explored, e.g., conditional random field (CRF) (Galley, 2006), Support Vector Regression (SVR) (Ouyang et al., 2011), and Logistic Regression (Li et al., 2013), etc.

Many query-focused summarizers are heuristic extensions of generic summarization methods by incorporating the information of the given query. A variety of query-dependent features were defined to measure the relevance, including TF-IDF cosine similarity (Wan and Xiao, 2009), WordNet similarity (Ouyang et al., 2011), and word co-occurrence (Prasad Pingali and Varma, 2007), etc. However, these features usually reward sentences similar to the query, which fail to meet the query need.

5.2 Deep Learning in Summarization

In the summarization area, the application of deep learning techniques has attracted more and more interest. (Genest et al., 2011) used unsupervised auto-encoders to represent both manual and system summaries for the task of summary evaluation. Their method, however, did not surpass ROUGE. Recently, some works (Cao et al., 2015a; Cao et al., 2015b) have tried to use neural networks to complement sentence ranking features. Although these models achieved the state-of-the-art performance, they still heavily relied on hand-crafted features. A few researches explored to directly measure similarity based on distributed representations. (Yin and Pei, 2015) trained a language model based on convolutional neural networks to project sentences onto distributed representations. (Cheng and Lapata, 2016) treated single document summarization as a sequence labeling task and modeled it by the recurrent neural networks. Others like (Kobayashi et al., 2015; Kågebäck et al., 2014) just used the sum of trained word embeddings to represent sentences or documents.

In addition to extractive summarization, deep learning technologies have also been applied to compressive and abstractive summarization. (Filippova et al., 2015) used word embeddings and Long Short Term Memory models (LSTMs) to output readable and informative sentence compressions. (Rush et al., 2015; Hu et al., 2015) leveraged the neural attention model (Bahdanau et al., 2014) in the machine translation area to generate one-sentence summaries. We have described these methods in Section 2.2.

6 Conclusion and Future Work

This paper proposes a novel query-focused summarization system called AttSum which jointly handles saliency ranking and relevance ranking. It automatically generates distributed representations for sentences as well as the document cluster. Meanwhile, it applies the attention mechanism that tries to simulate human attentive reading behavior when a query is given. We conduct extensive experiments on DUC query-focused summarization datasets. Using no hand-crafted features, AttSum achieves competitive performance. It is also observed that the sentences recognized to focus on the query indeed meet the query need.

Since we have obtained the semantic representations for the document cluster, we believe our system can be easily extended into abstractive summarization. The only additional step is to integrate a neural language model after document embeddings. We leave this as our future work.

Acknowledgements

The work described in this paper was supported by Research Grants Council of Hong Kong (PolyU 152094/14E, PolyU 152248/16E), National Natural Science Foundation of China (61272291 and 61672445) and The Hong Kong Polytechnic University (4-BCB5, B-Q46C and G-YBJP). The correspondence authors of this paper are Wenjie Li and Sujian Li.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015a. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of AAAI*.

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang. 2015b. Learning summary prior representation for extractive summarization. *Proceedings of ACL: Short Papers*, pages 829–833.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*.

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

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of DUC*, pages 1–12.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22(1):457–479.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of EMNLP*, pages 360–368.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of EMNLP*, pages 364–372.

Pierre-Etienne Genest, Fabrizio Gotti, and Yoshua Bengio. 2011. Deep learning for automatic summary scoring. In *Proceedings of the Workshop on Automatic Text Summarization*, pages 17–28.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on ILP for NLP*, pages 10–18.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In *Proceedings of EMNLP*, pages 1967–1972.

Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In *Proceedings of EACL Workshop*, pages 31–39.

Hayato Kobayashi, Masaki Noguchi, and Taichi Yatsuka. 2015. Summarization based on embedding distributions. In *Proceedings of EMNLP*, pages 1984–1989.

Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of ACL*, pages 1004–1013.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop*, pages 74–81.

Ryan McDonald. 2007. *A study of global inference algorithms in multi-document summarization*. Springer.

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

You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Information Processing & Management*, 47(2):227–237.

Karolina Owczarzak, John M Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9.

Rahul K Prasad Pingali and Vasudeva Varma. 2007. Iiit hyderabad at duc 2007. *Proceedings of DUC 2007*.

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

Xiaojun Wan and Jianguo Xiao. 2009. Graph-based multi-modality learning for topic-focused multi-document summarization. In *IJCAI*, pages 1586–1591.

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

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