Learning to Create Sentence Semantic Relation Graphs for Multi-Document Summarization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Facts can be expressed by a huge amount of sentences, which complicates the task of multi-document summarization. In this paper, we present *SemSentSum*, a fully data-driven model able to leverage sentence similarities by using universal sentence embeddings in order to build a sentence semantic relation graph and later, train specialized sentence embeddings for the task of multi-document summarization. We show that these two components are the keys of *SemSentSum* to achieve competitive results with other state-of-the-art models. We also devise a new greedy sentence selection method where sentence redundancies are computed using sentence embeddings. Finally, there is no need of hand-crafted features nor additional annotated data and therefore, *SemSentSum* can be used in real scenarios.

1 Introduction

Nowadays, information from the Web is overwhelming us from all sides and compels the need of automated multi-document summarization systems in order to produce high quality summaries. Extractive multi-document summarization, where the final summary is composed of sentences in the input documents, has been tackled by a large range of approaches. In the meanwhile, only an handful of researchers have studied abstractive multi-document summarization due to a lack of large datasets except the recent work of Liu et al. [2018], who create and use a large dataset based on Wikipedia (not yet public). Consequently, we only focus on extractive multi-document summarization.

Generally, summarization systems outputs summaries in two steps: sentence ranking followed by sentence selection. The first estimates an importance score of each sentence and the second chooses which ones to select by taking into account 1) their importance and 2) their redundancy among other sentences and the current summary. Most of the time, all sentences in the same collection of documents are processed independently and therefore, their relationships are lost. Due to the lack of data, models are generally forced to either heavily rely on well-designed features at the word level (Hong and Nenkova [2014], Cao et al. [2015], Christensen et al. [2013]) or take advantage of other large manually annotated data and then apply transfer learning (Cao et al. [2017]). In realistic scenarios, features are hard to craft and gathering additional annotated data is very expensive.

In this work, we present *SemSentSum*, a competitive fully data-driven model which does not depend on neither hand-crafted features nor additional data. It relies on the idea that sentence embeddings are able to capture the syntactic and semantic of sentences. We emphasize the fact that lot of ways exist to express a same sentence and unfortunately, this can not be handled by using only words as commonly done. Consequently, we first integrate universal sentence embeddings knowledge to build our sentence semantic relation graph so as to encapsulate sentence similarities. Secondly, in order to be related to the domains tackled by the collections of documents, we train specialized sentence embeddings by utilizing a sentence encoder in our model. Afterwards, we leverage the knowledge of our sentence semantic relation graph and the fine-grained sentence embeddings with a graph

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

convolutional network (Kipf and Welling [2017]). Finally, we employ a greedy strategy to produce summaries being informative and non-redundant by taking advantage of sentence embeddings to detect redundancies between candidate sentences and the current summary.

To the best of our knowledge, we are the first to leverage universal sentence embeddings in order to build a sentence relation graph and train more specialized sentence embeddings for the task of multi-document summarization. However, our method can be applied for other tasks such as information cascade, query-focused summarization, keyphrase extraction or information retrieval. In addition, we also are the first to propose a novel redundancy detection method based on sentence embedding while generating the final summary. Finally, we propose a fully data-driven model which does not need neither additional data nor hand-crafted features and is competitive with state-of-the-art systems.

2 Related Work

Extractive multi-document summarization has been tackled by a large range of approaches. On one hand, lot of graph-based methods exist. Radev [2000] introduces a cross-document structure theory as a basis to build multi-document summarization. Few years later, Erkan and Radev [2004a] propose LexRank, an unsupervised multi-document summarizer based on the concept of eigenvector centrality in a graph of sentences. Other works exploit shallow or deep features from the graph's topology (Mihalcea and Tarau, Wan and Yang [2006], Antiqueira et al. [2009]). Wan and Yang [2008] pairs graph-based methods (e.g. random walk) with clustering. Mei et al. [2010] improve results by using a reinforced random walk model to rank sentences and keep non-redundant ones. A novel system by Christensen et al. [2013] does sentence selection while balancing coherence and saliency. They build a graph which approximates discourse relations across sentences (Mann and Thompson [1988]).

On the other hand, different viable methods are available such as Maximum Marginal Relevance (Carbonell and Goldstein [1998]) using a greedy approach to select sentences and consider the tradeoff between relevance and redundancy, support vector regression (Li et al. [2007]), conditional random field (Galley [2006]), or hidden makov model (Conroy et al. [2004]). Some others rely on n-grams regression as in Hong and Nenkova [2014], Li et al. [2013], Conroy et al. [2006]. More recently, Cao et al. [2015] build a recursive neural network trying to find automatically combination of hand-crafted features. Cao et al. [2017] employ a neural model for text classification on a large manually annotated dataset and apply transfer learning for multi-document summarization afterwards. Surprisingly, neural networks work well for multi-document summarization even with small datasets.

The closest work to ours is Yasunaga et al. [2017]. They create a normalized version of the approximate discourse graph (Christensen et al. [2013]) where sentence nodes are normalized over all the incoming edges. Then, they feed a neural network composed of a sentence encoder, three graph convolutional layers, one document encoder and an attention mechanism. Afterwards, they greedily select sentences using tf-idf similarity to detect redundant sentences. However, our model differs in three ways: 1) we build our sentence semantic relation graph by using pre-trained sentence embeddings with cosine similarity and neither heavy preprocessing (besides tokenization) nor hand-crafted features. Thus our model is fully data-driven. 2) ***SemSentSum*** is a smaller model having similar sentence encoder and attention mechanism but less graph convolutional layers and no document encoder. 3) Our method for summary generation is also different as we leverage sentence embeddings to compute the redundancy between a candidate sentence and the current summary while Yasunaga et al. [2017], Hong and Nenkova [2014], Cao et al. [2015, 2017] utilize tf-idf approaches.

3 Method

Let C denote a collection of related documents composed of a set of documents $\{D_i | i \in [1, N]\}$ where N is the number of documents. Moreover, each document D_i consists of a set of sentences $\{S_{i,j} | j \in [1, M]\}$, M being the number of sentences in D_i. Given a collection of related documents C, our goal is to produce a summary Sum using a subset of these in the input documents ordered in some way, such that $Sum = (S_{i_1,j_1}, S_{i_2,j_2}, ..., S_{i_n,j_m})$.

In this section, we describe how our summarization system, called ***SemSentSum***, estimates the salience score of each sentence and how it selects a subset of these to create the final summary. The whole architecture of ***SemSentSum*** is depicted in Figure 1.

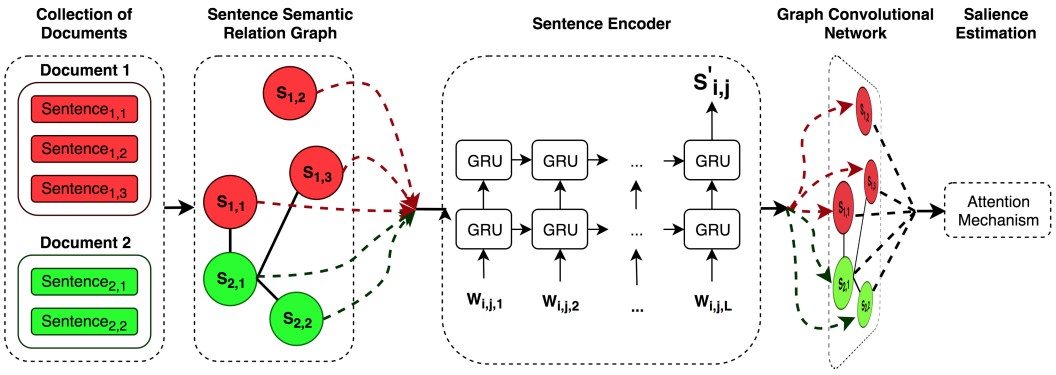

Figure 1: Overview of *SemSentSum*. This illustration includes two documents in the collection where the first one has three sentences and the second two. A sentence semantic relation graph is firstly built and each sentence node is processed by an encoder network thereafter. A graph convolutional network is applied on top and produces high-level hidden features for individual sentences. Finally, salience scores are estimated using an attention mechanism aligning sentences with cluster context.

In order to perform sentence selection, we first build our novel sentence semantic relation graph where each vertex is a sentence and edges capture the semantic similarity among them. Afterwards, each sentence is fed into a recurrent neural network, as sentence encoder, so as to generate sentence embeddings using the last hidden states. Thereafter, a graph convolutional neural network is applied on top where the sentence semantic relation graph is the adjacency matrix and the sentence embeddings are the node features. Then, the sentences are aligned with the cluster context via an attention mechanism in order to compute an estimate salience score representing how much salient is a sentence with respect to the final summary. Finally, based on the latter we devise an innovative greedy method, which leverages sentence embeddings to detect redundant sentences and select sentences until reaching the summary length limit.

3.1 Sentence Semantic Relation Graph

We model the semantic relationship among sentences using a graph representation. In this graph, each vertex is a sentence $S_{i,j}$ (j'th sentence of document D_i) from the collection documents C and an undirected edge between S_{i_u,j_u} and S_{i_v,j_v} indicates their degree of similarity. In order to compute the semantic similarity, we leverage novel sentence embeddings techniques by using the model of Pagliardini et al. [2018] trained on the English Wikipedia corpus. We process sentences by their model and compute the cosine similarity between every sentence pair. Having a complete graph does not allow the model to leverage much the semantic relation among sentences as every sentence pair is connected (i.e. the graph is complete). Furthermore, all edges have a similarity above zero as it is very unlikely that two sentence embeddings are completely orthogonal.

To overcome this problem, we hereby introduce an edge removal method which overcomes the aforementioned shortcoming: every edge below a certain threshold t^g_{sim} is removed in order to emphasize focused sentence similarity. Nonetheless, t^g_{sim} should not be too large otherwise the model will be prone to overfitting. After removing edges below t^g_{sim}, our sentence semantic relation graph is used as the adjacency matrix A for the graph convolutional network (see Section 3.3).

We hypothesize that incorporating general sentence embeddings into edges between sentences and later compute fine-grained sentence embeddings (see Section 3.2) for the goal of multi-document summarization is beneficial. As the pre-trained model uses English Wikipedia as source corpus, the corresponding texts are of high quality and therefore, allow to cover a lot of contexts for each encountered words as well as sentences. However, we still need more fine-grained sentence embeddings in order to be more related to the tackled domains of the collections of documents.

Finally, we highlight that 1) these pre-trained sentence embeddings are only used to compute the weights of the edges and will not be used later by the model (as others will be produced by the sentence encoder, see Section 3.2) 2) the edge weights are static and will not change during training.

3.2 Sentence Encoder

Given a list of documents C, we encode each document's sentence $S_{i,j}$, where each sentence has at most L words $(w_{i,j,1}, w_{i,j,2}, ..., w_{i,j,L})$. Every word is converted into word embeddings and then fed to the sentence encoder in order to compute sentence embeddings $S'_{i,j}$. We employ a two-layers forward recurrent neural network using Gated Recurrent Units (GRU) from Cho et al. [2014] as sentence encoder and the sentence embeddings are extracted from the last hidden states. We choose GRU instead of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber [1997]), due to their reduced number of parameters and their comparable performance (Chung et al. [2014]). We then concatenate all sentence embeddings into a matrix X which constitutes the input node features that will be used by the graph convolutional network (see Section 3.3).

3.3 Graph Convolutional Network

Once we have sentence embeddings and the sentence semantic relation graph, we apply a Graph Convolutional Network (GCN) from Kipf and Welling [2017] in order to capture high-level hidden features for each sentence, encapsulating sentence information as well as the graph-structure. We believe that that our sentence semantic relation graph contains information not present in the data and thus, we leverage this information by running graph convolutions.

The GCN model takes as input the node features matrix X containing all sentence embeddings of the collection of documents and a squared adjacency matrix A being our underlying sentence semantic relation graph. It outputs hidden representations for each node that encode both local graph structure and nodes' features. In order to take into account the sentences themselves during the information propagation, we add self-connections (i.e. the identity matrix) to A such that $\tilde{A} = A + I$.

Subsequently, we obtain our sentence hidden features by using Equation 1.

$$S''_{i,j} = f(X, A) = \text{ELU}(\tilde{A}\,\text{ELU}(\tilde{A}XW_g^{(0)} + b_g^{(0)})W_g^{(1)} + b_g^{(1)}) \tag{1}$$

where $W_g^{(i)}$ is the weight matrix of the i'th graph convolution layer and $b^{(i)}$ the bias vector. We choose Exponential Linear Unit (ELU) nonlinearity function from Klambauer et al. [2017] due to its ability to handle the vanishing gradient problem by making the mean activation close to zero and consequently, facilitates the backpropagation. By using only one hidden layer, as we only have one input-to-hidden layer and one hidden-to-output layer, we limit the information propagation to the first order neighborhood. Finally, GCN is a special form of Laplacian smoothing (Li et al. [2018]).

3.4 Saliency Estimation

We use an attention mechanism in order to estimate a salience score of each sentence. Given all sentence embeddings $S''i, j$, we compute a context embeddings so as to have a global view of the whole collection of documents. For this purpose, we take the mean of all sentence embeddings in the current documents using Equation 2.

$$S^C = \frac{1}{M \cdot N} \sum_{i=0}^{N} \sum_{j=0}^{M} S''i, j \tag{2}$$

Afterwards, we align the sentence embeddings with the context embeddings using Equation 3, similarly to Bahdanau et al. [2015], Vinyals et al. [2015], Yasunaga et al. [2017].

$$f(S''_{i,j}) = v_a^T \text{tanh}(W_a^{(0)} S''i, j + W_a^{(1)} S^C) \tag{3}$$

We then normalize the scores via softmax and obtain our estimated salience score $S^s_{i,j}$.

3.5 Training

Our model *SemSentSum* is trained end-to-end to minimize the cross-entropy loss of Equation 4 between the salience score prediction and the normalized ROUGE score for each sentence, as in Cao et al. [2015], Yasunaga et al. [2017]. The learnable parameters of *SemSentSum* are these of the sentence encoder and $W_g^{(0)}, b_g^{(0)}, W_g^{(1)}, b_g^{(1)}, W_a^{(0)}, W_a^{(1)}$ and v_a as well.

Table 1: Statistics on the DUC datasets for multi-document summarization.

Year	#Clusters	#Documents	#Sentences	Summary Length Limit
2001	30	309	11295	100 Words
2002	59	567	15878	100 Words
2003	30	298	7721	100 Words
2004	50	500	13280	665 Bytes

$$\mathcal{L} = -\sum_{C}\sum_{D \in C}\sum_{S \in D} R(S) \log S^s \tag{4}$$

$R(S)$ is computed as being the average between ROUGE-1 and ROUGE-2 recall scores (see Section 4.2 for more information about ROUGE scores), following the common practice in the area of single and multi-document summarization. Furthermore, we normalize the ROUGE scores with a rescaling factor α to make the distribution sharper, as in Yasunaga et al. [2017]. The intuition is that the scale of raw ROUGE scores is not necessarily good for a softmax normalization.

3.6 Summary Generation Process

While our model *SemSentSum* provides estimated saliency scores, we employ a simple innovative greedy strategy to offer a summary Sum being informative and non-redundant. We first discard sentences having less than 9 words, as in Erkan and Radev [2004b], and then sort them in descending order of their estimated salience scores. To create our final summary Sum, we iteratively dequeue the sentence having the highest score and append it to the current summary if it is non-redundant with respect to Sum. We iterate until reaching the summary length limit.

We introduce a novel way to determine the redundancy of a candidate sentence with the current summary: a sentence is considered as non-redundant if and only if the cosine similarity between its sentence embeddings and the embeddings of the current summary is below a certain threshold t^s_{sim}. As in Section 3.3, we use the pre-trained model of Pagliardini et al. [2018] to compute sentence as well as summary embeddings. We innovate by focusing on the semantic instead of word similarity like tf-idf approaches as in Hong and Nenkova [2014], Cao et al. [2015], Yasunaga et al. [2017], Cao et al. [2017], which might not reflect meaning similarity.

4 Experiments

4.1 Datasets

Experiments are conducted on the most commonly used datasets for multi-document summarization from the Document Understanding Conferences (DUC).[1]. We use DUC 2001, 2002, 2003 and 2004 as the tasks of generic multi-document summarization because they have been carried out during these years. We use DUC 2001 and 2002 for training, DUC 2003 for validation and finally, DUC 2004 for testing as the common practice. The Table 1 shows the number of clusters, documents, sentences and summary length limit for each of the dataset.

4.2 Evaluation Metric

For the evaluation, we use ROUGE (Lin [2004]) with the official parameters of DUC tasks and also truncate the summaries to 100 words for DUC 2001/2002/2003 and to 665 bytes for DUC 2004 (if not explicitly stated otherwise).[2] Notably, we take ROUGE-1 and ROUGE-2 recall scores (in percent) as the main metrics for comparison between produced summaries and golden ones as proposed by Owczarzak et al. [2012]. The goal of the ROUGE-N metric is to compute the ratio of the number of N-grams from the generated summary matching these of the human reference summaries.

[1] https://www-nlpir.nist.gov/projects/duc/guidelines.html

[2] ROUGE-1.5.5 with options: -n 2 -m -u -c 95 -x -r 1000 -f A -p 0.5 -t 0 and -l 100 if using DUC 2001/2002/2003 otherwise -b 665.

4.3 Model Settings

To define the edge weights of our sentence semantic relation graph, we employ the 600-dimensional pre-trained unigram model of Pagliardini et al. [2018] using English Wikipedia as source corpus. We keep only edges having a weight larger than $t^g_{sim} = 0.5$. For word embeddings, the 300-dimensional pre-trained GloVe embeddings (Pennington et al. [2014]) are used and not tuned during training. The output dimension of the sentence embeddings produced by the sentence encoder is the same as these of the word embeddings, i.e. 300. For the graph convolutional network, the number of hidden units is 32 and the size of the generated hidden feature vectors is also 300. The rescaling factor to make the ROUGE distribution sharper in the loss function is $\alpha = 40$ (tuned on the validation set). We use a batch size of 1, a learning rate of 0.001 using Adam (Kingma and Ba [2015]) as optimizer. In order to make *SemSentSum* generalize better, we use dropout (Srivastava et al. [2014]) of 0.4, clip the gradient norm at 1.0 if higher, add L2-norm regularizer with a regularization factor of 10^{-11} and train using early stopping with a patience of 10 iterations. Finally, the redundancy threshold t^s_{sim} in the summary generation process is 0.8. We train on a GeForce Titan Xp GPU in a couple of minutes.

4.4 Summarization Performance

We asset the performance of our model *SemSentSum* by training it on DUC 2001/2002, tuning it on DUC 2003 and evaluating it on DUC 2004. In order to fairly compare *SemSentSum* with more models available in the literature, experiments are conducted with summaries truncated to 665 bytes (official parameters of the DUC competition) but also with summaries with a length constraint of 100 words.

4.5 Sentence Semantic Relation Graph Generation

We investigate miscellaneous methods to build the sentence semantic relation graph and vary the value of t^g_{sim} from 0.0 to 0.75 to study the impact of the threshold cut-off. Among these:

1. *Cosine*: Using cosine similarity as in Section 3.1;
2. *tf-idf*: Considering a node as the query and another as document. The weight corresponds to the similarity between the query and the document;
3. *TextRank* (Mihalcea and Tarau [2004]): A weighted graph is created where nodes are sentences and edges defined by a similarity measure based on word overlap. Afterwards, an algorithm similar to PageRank (Page et al. [1998]) is used to compute sentence importance and refined edge weights;
4. *LexRank* (Erkan and Radev [2004a]): An unsupervised multi-document summarizer based on the concept of eigenvector centrality in a graph of sentences to set up the edge weights;
5. *Approximate Discourse Graph* (ADG) (Christensen et al. [2013]): Approximation of a discourse graph where nodes are sentences and an edge (S_u, S_v) indicates the sentence S_v can be placed right after S_u in a coherent summary;
6. *Personalized ADG* (PADG) (Yasunaga et al. [2017]): Normalized version of ADG where sentence nodes are normalized over all the incoming edges.

4.6 Ablation Study

In order to quantify the contribution of the different components of *SemSentSum*, we try variations of our model by removing different modules one at a time. Our three main elements are: the sentence encoder (*Sent*), the graph convolutional neural network (*GCN*) and the attention mechanism (*Att*). When we omit *Sent*, we substitute it with the pre-trained sentence embeddings used in Section 3.1.

4.7 Results and Discussion

Three axis are used to evaluate our model *SemSentSum*: 1) the summarization performance to asset the capability of our proposed model 2) the impact of the sentence semantic relation graph generation using various methods and different thresholds t^g_{sim} 3) an ablation study to analyze the importance of each component of *SemSentSum*.

Table 2: Comparison of *SemSentSum* with various models published in the literature. Evaluation done using ROUGE-1/ROUGE-2 recall scores (%) on DUC 2004 with 665 bytes/100 words summaries.

	665 bytes summaries		100 words summaries	
Model	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
MMR (Bennani-Smires et al. [2018])	35.49	7.50		
PV-DBOW+BS (Mani et al. [2017])	36.10	6.77		
SVR (Li et al. [2007])	36.18	9.34		
G-Flow (Christensen et al. [2013])	37.38	8.77	35.30	8.27
R2N2 (Cao et al. [2015])	38.16	9.52		
TCSum (Cao et al. [2017])	38.27	9.66		
SemSentSum (Our model)	37.69	9.10	38.46	9.34
FreqSum (Nenkova et al. [2006])			35.30	8.11
Cont. LexRank (Erkan and Radev [2004b])			35.95	7.47
Centroid (Radev et al. [2004])			36.41	7.97
CLASSY11 (Conroy et al. [2011])			37.22	9.20
CLASSY04 (Conroy et al. [2004])			37.62	8.96
GreedyKL (Haghighi and Vanderwende [2009])			37.98	8.53
GRU+GCN+PADG (Yasunaga et al. [2017])			38.23	9.48
RegSum (Hong and Nenkova [2014])			38.57	9.75

Summarization Performance All the results are reported in Table 2. Firstly, if we compare *SemSentSum* with summary length of 665 bytes, we note that our model largely outperforms the baselines relying on sentence embeddings (Bennani-Smires et al. [2018]) or document embeddings (Mani et al. [2017]). In addition, *SemSentSum* also beats a graph-based method based on approximating discourse graph (Christensen et al. [2013]). Then comes the recursive neural networks of Cao et al. [2015] learning automatically combinations of hand-crafted features, therefore relying heavily on these. The last model to compare with is the model of Cao et al. [2017] using transfer learning from a text classifier model based on a domain-related dataset of $30'000$ documents (from New York Times having same tackled topics). Consequently, both models perform better because of the use of either rich set of hand-crafted features or an extra dataset 30 times larger than the DUC ones. However, we emphasize that our model is fully data-driven and do not rely on 1) hand-crafted features 2) extra large manually annotated dataset and as a results, 3) is usable in real scenarios.

Comparing with models producing longer summaries, i.e. 100 words, *SemSentSum* outperforms commonly used baselines (Nenkova et al. [2006]) and traditional graph-based approaches such as Radev et al. [2004], Erkan and Radev [2004b], Christensen et al. [2013]. As can be seen, the best model in the DUC competition (Conroy et al. [2004]), its improved version (Conroy et al. [2011]) and the greedy model of Haghighi and Vanderwende [2009] are underperforming compared to *SemSentSum*. The model of Yasunaga et al. [2017] relies on hand-crafted features to build the approximate discourse graph followed by a sentence encoder, three layers of graph convolutional networks, a document encoder and finally an attention mechanism. We perform better in term of ROUGE-1 score whereas slightly lower for ROUGE-2. However, our model is still competitive as 1) does not rely on approximate discourse graph being heavy to build in term of preprocessing 2) is much smaller because it depends on less layers of graph convolutional networks and does not need a document encoder. Finally, the model *RegSum* (Hong and Nenkova [2014]) computes sentence saliences based on word scores, incorporating rich set of word-level features. Nonetheless, our model is still competitive and does not depend on hand-crafted features due to its full data-driven nature.

Sentence Semantic Relation Graph The Table 3 shows the results of different methods to create the sentence semantic relation graph with various threshold t^g_{sim}. A first observation is that using cosine similarity with sentence embeddings significantly outperforms all other methods for ROUGE-1 and ROUGE-2 scores, mainly because it relies on the semantic of sentences instead of their individual words. A second is that different methods evolve similarly: *PADG, Lexrank, tfidf* behave similarly to an U-shaped curve whereas *Textrank* and *ADG* seem to perform better while increasing thresholds. Finally, the cosine method is the only one following an inverted U-shaped curve. The reason behind this behavior is in consequence of its distribution being similar to a normal distribution because it relies on the semantic instead of words, while the others are more skewed towards zero.

Table 3: ROUGE-1 and ROUGE-2 recall scores (%) for various methods to build the sentence semantic relation graph with different thresholds t^g_{sim}, then run on top of it *SemSentSum*.

		ROUGE-1					ROUGE-2			
Method	t^g_{sim}	0.0	0.25	0.5	0.75	t^g_{sim}	0.0	0.25	0.5	0.75
cosine		36.96	37.38	37.69	34.09		8.62	8.68	9.10	6.52
tf-idf		33.97	33.62	33.18	33.65		6.48	6.88	6.07	6.10
Textrank		32.75	32.69	33.67	33.81		6.23	6.27	6.40	6.42
Lexrank		35.17	34.59	33.87	33.97		8.00	7.52	6.37	6.68
ADG		32.90	33.71	33.95	34.02		6.02	6.40	6.33	6.69
PADG		34.25	33.37	33.80	33.95		6.86	6.12	6.42	6.42

Table 4: Ablation test performance. *Sent* corresponds to the sentence encoder, *Att* the attention mechanism and *GCN* the graph convolutional network.

Model	ROUGE-1	ROUGE-2
SemSentSum	37.69	9.10
- w/o Att	36.25	8.37
- w/o GCN	34.03	6.80
- w/o Sent	34.37	7.55
- w/o Att,GCN	30.83	4.74
- w/o Att,Sent	28.69	4.37
- w/o GCN,Sent	32.60	5.77
- w/o Att,GCN,Sent	28.93	4.18

Ablation Study We quantify the contribution of each module of *SemSentSum* in Table 4. Horizontal lines separate the number of components removed at the same time. By removing only one module, we observe that the drastic drop in term of performance is achieved when the graph convolutional network component is disabled. This emphasizes that relationship between sentences is indeed important and not present in the data itself. Moreover, this shows that our sentence semantic relation graph is able to capture sentence similarities by analyzing the semantic. Similarly, by removing the sentence encoder, another major decrease is noted, showing that using only universal sentence embeddings is not enough for the task of multi-document summarization and therefore, we need to leverage a more fine-tuned version for the domains tackled in DUC corpus. Finally, the attention mechanism seems to boost the capability of *SemSentSum* with embeddings aligned with the context.

As previously, removing the sentence encoder or the graph convolutional network in addition to the attention mechanism is still the most harmful, probably for the same reasons. However, an interesting point is by only letting the attention mechanism enabled, the model is still able to learn something in spite of having much worse results. Finally, when removing all components apart from the pre-trained sentence embeddings as only features, the model is unsurprisingly not learning and perform the worst.

5 Conclusion

In this work, we introduce a fully data-driven model *SentSum* not using neither hand-crafted features nor additional annotated data while being competitive with the state-of-the-art multi-document summarization systems. *SentSum* leverages universal sentence embeddings so as to create a sentence semantic relation graph and trains in the meantime more specialized sentence embeddings. It allows to capture sentence semantic and similarities whereas this is not possible by using only words as commonly done. We show that these elements are the key of the success of *SentSum*. In realistic scenarios, efficient hand-crafted features are cumbersome and additional annotated data is very costly to gather while sentence embeddings are easy to obtain and fast to produce. Moreover, we innovate by using sentence embeddings to compute redundancies between candidate sentences and the summary.

We believe that our sentence semantic relation graph and our model can be used for other tasks including information cascade, query-focused summarization, keyphrase extraction or information retrieval etc. In addition, we plan to let the weights of the sentence semantic relation graph be dynamic during training and also introduce attention mechanism to put more focus on certain sentences.

References

Lucas Antiqueira, Osvaldo N. Oliveira, Luciano da Fontoura Costa, and Maria das Graças Volpe Nunes. A complex network approach to text summarization. *Information Sciences*, 179 (5):584 – 599, 2009. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2008.10.032. URL http://www.sciencedirect.com/science/article/pii/S0020025508004520. Special Section - Quantum Structures: Theory and Applications.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*, September 2015. URL https://arxiv.org/abs/1409.0473.

Kamil Bennani-Smires, Claudiu Musat, Martin Jaggi, Andreea Hossmann, and Michael Baeriswyl. Embedrank: Unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*, 2018.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2153–2159. AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2886521.2886620.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3053–3059, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14525.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.291025. URL http://doi.acm.org/10.1145/290941.291025.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1179.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1173, 2013.

Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning Workshop, 2014*, 2014. URL https://arxiv.org/abs/1412.3555. Presented at the Deep Learning workshop at NIPS2014.

John M Conroy, Judith D Schlesinger, Jade Goldstein, and Dianne P O'leary. Left-brain/right-brain multi-document summarization. In *Proceedings of the Document Understanding Conference (DUC 2004)*, 2004.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 152–159, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1273073.1273093.

John M Conroy, Judith D Schlesinger, Jeff Kubina, Peter A Rankel, and Dianne P O'Leary. Classy 2011 at tac: Guided and multi-lingual summaries and evaluation metrics. *TAC*, 11:1–8, 2011.

Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004a. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622487.1622501.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004b.

Michel Galley. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 364–372, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-73-6. URL http://dl.acm.org/citation.cfm?id=1610075.1610126.

Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 362–370, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL http://dl.acm.org/citation.cfm?id=1620754.1620807.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, 2014.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017. URL http://arxiv.org/abs/1609.02907.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 972–981, 2017. URL http://papers.nips.cc/paper/6698-self-normalizing-neural-networks.

Chen Li, Xian Qian, and Yang Liu. Using supervised bigram-based ilp for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1004–1013, 2013.

Q. Li, Z. Han, and X.-M. Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *The Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, 2018.

Sujian Li, You Ouyang, Wei Wang, and Bin Sun. Multi-document summarization using support vector regression. In *In Proceedings of DUC*. Citeseer, 2007.

C. Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS)*, Barcelona, Spain, July 25-26 2004.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *International Conference on Learning Representations*, 2018. URL http://arxiv.org/abs/1801.10198.

Kaustubh Mani, Ishan Verma, and Lipika Dey. Multi-document summarization using distributed bag-of-words model. *arXiv preprint arXiv:1710.02745*, 2017.

William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: The interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1009–1018, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835931. URL http://doi.acm.org/10.1145/1835804.1835931.

R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004.

Rada Mihalcea and Paul Tarau. A language independent algorithm for single and multiple document summarization. In *In Proceedings of IJCNLP'2005*.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 573–580, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148269. URL `http://doi.acm.org/10.1145/1148170.1148269`.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 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, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2391258.2391259`.

L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998. URL `citeseer.nj.nec.com/page98pagerank.html`.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Dragomir R. Radev. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue - Volume 10*, SIGDIAL '00, pages 74–83, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1117736.1117745. URL `https://doi.org/10.3115/1117736.1117745`.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1): 1929–1958, January 2014. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=2627435.2670313`.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

Xiaojun Wan and Jianwu Yang. Improved affinity graph based multi-document summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 181–184, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1614049.1614095`.

Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 299–306, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390386. URL `http://doi.acm.org/10.1145/1390334.1390386`.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev. Graph-based neural multi-document summarization. In *Proceedings of CoNLL-2017*. Association for Computational Linguistics, 2017.