Abstractive Unsupervised Multi-Document Summarization using Paraphrastic Sentence Fusion

Mir Tafseer Nayeem
University of Lethbridge
Lethbridge, AB, Canada
mir.nayeem@uleth.ca

Tanvir Ahmed Fuad
University of Lethbridge
Lethbridge, AB, Canada
t.fuad@uleth.ca

Yllias Chali
University of Lethbridge
Lethbridge, AB, Canada
chali@cs.uleth.ca

Abstract

In this work, we aim at developing an unsupervised abstractive summarization system in the multi-document setting. We design a paraphrastic sentence fusion model which jointly performs sentence fusion and paraphrasing using skip-gram word embedding model at the sentence level. Our model improves the information coverage and at the same time abstractiveness of the generated sentences. We conduct our experiments on the human-generated multi-sentence compression datasets and evaluate our system on several newly proposed Machine Translation (**MT**) evaluation metrics. Furthermore, we apply our sentence level model to implement an abstractive multi-document summarization system where documents usually contain a related set of sentences. We also propose an optimal solution for the classical summary length limit problem which was not addressed in the past research. For the document level summary, we conduct experiments on the datasets of two different domains (e.g., news article and user reviews) which are well suited for multi-document abstractive summarization. Our experiments demonstrate that the methods bring significant improvements over the state-of-the-art methods.

1 Introduction

The task of automatic document summarization aims at finding the most relevant informations in a text and presenting them in a condensed form. A good summary should retain the most important contents of the original document or a cluster of related documents, while being coherent, non-redundant and grammatically readable. There are two types of summarizations: abstractive summarization and extractive summarization. Abstractive methods need extensive natural language generation to rewrite the sentences (Chali et al., 2017). Therefore, research community is focusing more on extractive summaries, which selects salient (important) sentences from the source document without any modification to create a summary. The abstractive techniques which are traditionally used are sentence compression, syntactic reorganization and lexical paraphrasing. Summarization is classified as single-document or multi-document based upon the number of source document. The information overlap between the documents from the same topic makes the multi-document summarization more challenging than the task of summarizing single documents. However, in case of multi-document summarization where source documents usually contain similar information, the extractive methods would produce redundant summary or biased towards specific source document (Nayeem and Chali, 2017a).

Multi-sentence compression (**MSC**) can be a useful solution for the above problem. It usually takes a group of related sentences and produces an output sentence through merging the sentences about the same topic, retaining the most important information and still maintain the grammaticality of the generated sentence. MSC is a text-to-text generation process in which a novel sentence is produced as a result of summarizing a set of similar sentences originally called sentence fusion (Barzilay and McKeown, 2005). On the other hand, lexical paraphrasing aims at replacing some selected words with other similar words while preserving the meaning of the original text. A good lexical substitution for a target word needs to be semantically similar to the target word and compatible with the given context (Melamud et al., 2015).

Proceedings of the 27th International Conference on Computational Linguistics, pages 1191–1204
Santa Fe, New Mexico, USA, August 20-26, 2018.

For example, the sentence "Jack composed these verses in 1995" could be lexically paraphrased into "Jack wrote these lines in 1995" without altering the sense of the initial sentence. The contributions of this paper are as follows:

- We design a novel abstractive sentence generation model which jointly performs sentence fusion and paraphrasing using skipgram word embedding model.
- We apply our sentence level model to design a full abstractive multi-document summarization system and achieved the state-of-the-art results on two different datasets. Different from the recent neural abstractive models, our model is completely unsupervised, full abstractive (not limited to deletion based compressions) and applied to multi-document summarization.
- We also propose an optimal solution for the classical summary length limit problem in multi-document setting.

2 Related Work

Abstractive summarization is generally much more difficult which involves sophisticated techniques for meaning representation, content organization, sentence compression, sentence fusion, paraphrasing etc. There has been a huge interest on compressive document summarization that tries to compress original sentences to form a summary (Clarke and Lapata, 2006; Clarke and Lapata, 2008; Filippova, 2010) as a first intermediate step towards abstractive summarization. Compressive summarization techniques include sentences which are compressed from original sentences without further modifications other than word deletion. Sentence compression involving two or more sentences is called **MSC** (Multi-Sentence Compression). Most of the previous MSC approaches rely on the syntactic parsing to build the dependency tree for each related sentence in a cluster for producing grammatical compressions (Filippova and Strube, 2008). Unfortunately, syntactic parsers are not available for all the languages. As an alternative, word graph-based approaches that only require a POS tagger and a list of stopwords have been proposed first by (Filippova, 2010). A directed word graph is constructed in which nodes represent words and edges represent the adjacency between words in a sentence. Hence, compressed sentences are generated by finding k-shortest paths in the word graph. (Boudin and Morin, 2013) improved Filippova's approach by re-ranking the fusion candidate paths according to keyphrases to generate more informative sentences. However, grammaticality is sacrificed to improve informativity in these works (Nayeem and Chali, 2017b).

(Banerjee et al., 2015) proposed an abstractive multi-document summarization system using sentence fusion approach of (Filippova, 2010) combined with Integer Linear Programming (**ILP**) sentence selection. Following (Banerjee et al., 2015) work, several recent approaches have been proposed with slight modifications. Multiword Expressions (**MWE**) was exploited in (ShafieiBavani et al., 2016) to produce more informative compressions. Recently, (Tuan et al., 2017) include syntax factor along with (Banerjee et al., 2015) to improve performance. However, all the above mentioned systems try to produce compressions by copying the source sentence words, no paraphrasing is involved in the process.

Recently end-to-end training with encoder-decoder neural networks have achieved huge success in case of abstractive summarization. These systems have adopted techniques such as encoder-decoder with attention (Bahdanau et al., 2015; Luong et al., 2015) neural network models from the field of machine translation to model the sentence summarization task. (Rush et al., 2015) was the first to use neural sequence-to-sequence learning in headline generation task from a single document. Unfortunately, this line of research under the term sentence summarization (Rush et al., 2015), which can generate only a single sentence, somewhat misleadingly called text summarization in some follow-up research works (Nallapati et al., 2016; Chopra et al., 2016; Suzuki and Nagata, 2017; Zhou et al., 2017; Ma et al., 2017). There are some limitations to the above mentioned models, one is that the the produced output is also very short (about 75 characters). Same as the headlines, their model produces ungrammatical sentences during generation. However, there are some recent attempts which uses **CNN/DailyMail** corpus (Hermann et al., 2015) as a supervised training data to generate multi-sentence summary from a single document (See

et al., 2017; Li et al., 2017b; Paulus et al., 2017; Narayan et al., 2018a; Narayan et al., 2018b). The recent abstractive summarization models actually produce compressive summaries by deleting the words form a single source document, no direct paraphrasing was involved in the process. Hence, no new words were generated which are different form the source document words (other than morphological variation), which is pointed out by their own experimental results. Very recently, some researchers employ neural network based framework to tackle the summarization problem in multi-document setting (Yasunaga et al., 2017; Li et al., 2017a). (Yasunaga et al., 2017) is limited to extractive summmarization. On the other hand, (Li et al., 2017a) is limited to compressive summary generation using an ILP based model, and there is no explicit redundancy control in the summary side. Unfortunately, full abstractive summarization in multi-document setting still lacks satisfactory solutions due to the lack of large multi-document summarization datasets needed to train the computationally expensive sequence-to-sequence models. In this paper, we tackle this issue in an unsupervised way using deep representation learning.

3 Paraphrastic Sentence Fusion Model

Most of the previous works rely only on deletion based compressions, either sentence compression or fusion for abstracting sentences. Instead, in this paper we take the first step towards finding a joint representation for sentence abstraction using sentence fusion and lexical paraphrase rather than treating these two independently.

3.1 Word Graph Construction for Sentence Fusion

Given a cluster of related sentences we construct a word-graph following (Filippova, 2010; Boudin and Morin, 2013). Let $S = \{s_1, s_2, ..., s_n\}$ be a set of related sentences, we construct a graph $G = (V, E)$ by iteratively adding sentences to it. The vertices are the words along with the parts-of-speech (POS) tags and directed edges are formed by simply connecting the adjacent words in the sentences. Once the first sentence is added, words from the other related sentences are mapped onto a node in the graph provided that they have exactly the same lower cased word form and the same POS tag. Each sentence is connected to dummy start and end nodes to mark the beginning and ending of the sentences. Figure 1 illustrates an example word-graph for the following two sentences,

S1: In Asia Japan Nikkei lost 9.6% while Hong Kongs Hang Seng index fell 8.3%.

S2: Elsewhere in Asia Hong Kongs Hang Seng index fell 8.3% to 12,618.

As we can see, the two input sentences contain similar information, but differs in sentence length, syntax, and the detail of information. The solid directed arrows connect the words in the first sentence S1, while the dotted arrows join the words in the second sentence S2. After constructing the word-graph using (Filippova, 2010; Boudin and Morin, 2013) as described above, we can generate K-shortest paths from dummy start node to end node in the word graph (see Figure 1). For example, we can generate these paths:

Ex1: In Asia Hong Kongs Hang Seng index fell 8.3%.

Ex2: Elsewhere in Asia Hong Kongs Hang Seng index fell 8.3%.

Ex3: Elsewhere in Asia Japan Nikkei lost 9.6% while Hong Kongs Hang Seng index fell 8.3%.

The above examples are sampled from the K-shortest paths generated from the word-graph G (K is usually ranges from 50 to 200 according to the literature (Filippova, 2010; Boudin and Morin, 2013)). The main challenge is to rank these K fused sentences according to the information they contain. Hence, we design a candidate ranking strategy to sort the generated K-shortest paths based on the information coverage.

3.2 Candidate Ranking

We rank the fused candidates by applying **TextRank** algorithm (Mihalcea and Tarau, 2004) which involves constructing an undirected graph where candidates are vertices, and weighted edges are formed by connecting candidate sentences by a similarity metric. Original TextRank algorithm determines the similarity based on the lexical overlap. However, this algorithm has a serious drawback: If two sentences

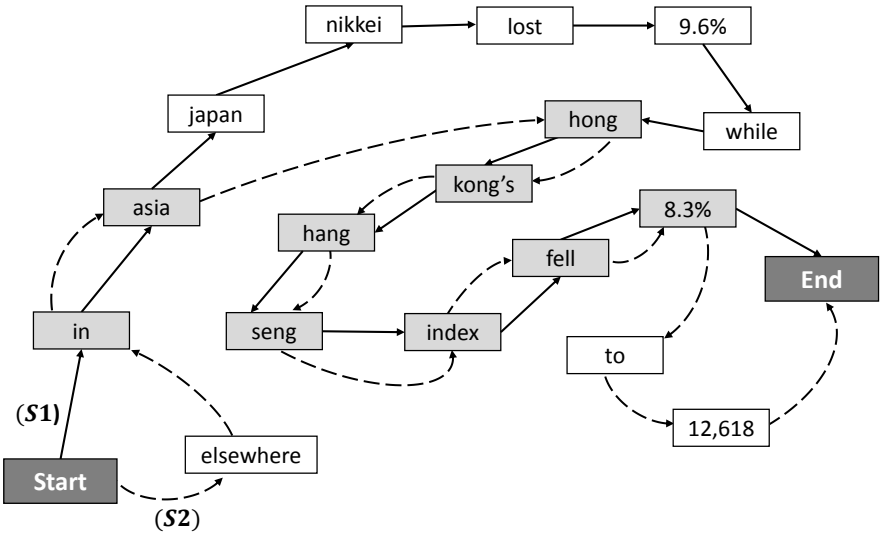

Figure 1: Constructed Word graph and a possible compression path (light gray nodes)

are talking about the same topic without using any overlapped words, there will be no edge between them. Instead, we apply the following representation of a sentence to capture the semantic information.

3.2.1 Sentence Embedding

A sentence is a sequence of words $\mathbf{S} = (w_1, w_2, ..., w_L)$, where L is the length of the sentence. We encode a sentence using bi-directional GRUs (Cho et al., 2014). In the simplest uni-directional case, while reading input symbols from left to right, a GRU learns the hidden annotations h_t at time t with

$$h_t = \mathbf{GRU}(h_{t-1}, e(w_t)) \tag{1}$$

where, the $h_t \in \mathbb{R}^n$ encodes all content seen so far at time t which is computed from h_{t-1} and $e(w_t)$, where $e(w_t) \in \mathbb{R}^m$ is the m-dimensional embedding of the current word w_t. We use 300-dimensional pre-trained word2vec embeddings[1](Mikolov et al., 2013) for each word as input to GRU.

As shown in Figure 2, **Bi-GRU** processes the input sentence in both forward and backward direction with two separate hidden layers calculated with GRUs, obtains the forward hidden states $(\overrightarrow{h_1}, \ldots, \overrightarrow{h_L})$ and the backward hidden states $(\overleftarrow{h_1}, \ldots, \overleftarrow{h_L})$. For each position t, we simply concatenate both forward and backward states into the final hidden state:

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{2}$$

in which operator $\oplus$ indicates concatenation. $\overrightarrow{h_t}$ is calculated using Eq. (1) and $\overleftarrow{h_t}$ is calculated using the following equation.

$$\overleftarrow{h_t} = \mathbf{GRU}(\overleftarrow{h_{t+1}}, e(w_t)) \tag{3}$$

$\overrightarrow{h_0}$ is initialized as zero vector, and the output sentence embedding x_i for the sentence S_i is the last hidden state:

$$x_i = h_L \tag{4}$$

We start by constructing an undirected graph where fused sentence candidates are vertices, and weighted edges are formed by measuring the cosine distance between the candidate sentence embeddings obtained from equation (4). After we have our graph, we run the TextRank (Mihalcea and Tarau,

[1]https://code.google.com/archive/p/word2vec/

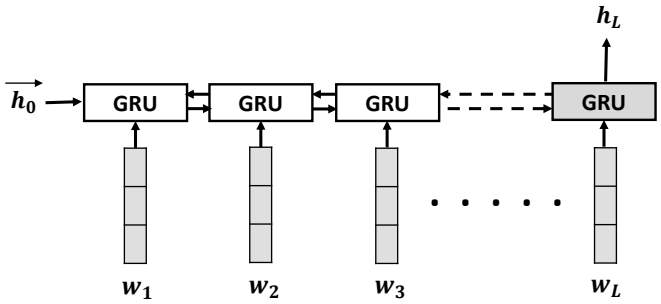

Figure 2: Sentence Embedding

2004) algorithm on it. This involves initializing a score of 1 for each vertex, and repeatedly applying the TextRank update rule until convergence. After reaching convergence, we extract the fused candidate sentences along with TextRank scores. For instance, $Rank(S_i)$ indicates the importance score assigned to sentence S_i.

3.3 Context Sensitive Lexical Substitution

3.3.1 Target Word Identification for Substitution

After constructing the word-graph as presented in section 3.1, we take only the nouns and verbs for possible substitution candidates from the word-graph G. We didn't consider the named entities, where, $NE \in \{PER; LOC; ORG; MISC\}$ for the substitution.

3.3.2 Substitution Selection

The **PPDB 2.0** (Pavlick et al., 2015) provides millions of lexical, phrasal and syntactic paraphrases which come into packages of different sizes (going from S to XXXL). For our model, we use the lexical XXL. For instance, we can gather lexical substitution set S = {*gliding, sailing, diving, travelling*} for the target word ($t = flying$) from PPDB 2.0. We hardcoded the model to select substitutes with the same POS tag and that are not a morphological variant (*e.g., fly, flew, flown*).

3.3.3 Substitution Ranking

Word embeddings are low-dimensional vector representations of words such as **word2vec** (Mikolov et al., 2013) that recently gained much attention in various semantic tasks. **Word2vecf** (Levy and Goldberg, 2014) is an extension of word2vec to produce syntax-based word embeddings. They show that these embeddings tend to capture functional word similarity (as in *manage* $\rightarrow$ *supervise*) rather than topical similarity (as in *manage* $\rightarrow$ *manager*). We use the word and context vectors released by (Melamud et al., 2015) which was shown to perform strongly on lexical substitution task. These embeddings contain 600d (600 dimension) vectors for 173k words and about 1M syntactic contexts processed using the dependency based word2vecf model (Levy and Goldberg, 2014). Their measure $addCos$ for estimating the appropriateness of a substitute s from the substitution set S, for the target word t in the set of the target word's context elements $C = \{c_1, c_2, ..., c_n\}$, which is defined as follows,

$$addCos(s|t, C) = \frac{cos(s, t) + \sum_{c \in C} cos(s, c)}{|C| + 1}$$

Finally, we select the best substitution $\mathbf{s}$ according to maximum $\mathbf{addCos}$ scores over $\mathbf{0.7}$ and attach it with the target word vertex $\mathbf{t}$ in the word-graph G along with the $\mathbf{addCos}$ score. For the other vertices which don't have substitution alternatives, we assign an $\mathbf{addCos}$ score of zero in the word-graph G.

3.3.4 Confidence Score

Once the substitutions are placed into the word-graph G, in order to maintain the grammaticality and the reliability of the final generated sentences, we use a 3-gram language model, which assigns probabilities to sequence of words in a generated candidate. Suppose that a candidate contains a sequence of m

words $\{w_1, w_2, w_3, \ldots, w_m\}$. The score **CS** (Confidence Score) assigned to each candidate is defined as follows:

$$CS(w_1, ..., w_m) = \frac{1}{1 - Score_{LM}(w_1, ..., w_m)}$$

In our experiment, we used a language model that is trained on the English Gigaword corpus[2]. In the word-graph G, for each substitution candidate we calculate the confidence score with the adjacent vertices (in our case words) and calculate their average. We also assign this confidence score with the substitution vertex in the word-graph G.

Finally, we rank the K candidate fusions and find the N-best paraphrastic sentence fusion which balances the information coverage and the abstractiveness. The score of a candidate sentence fusion c is given by the following linear combination between the candidate rank score and the abstractiveness score (where, we set $\alpha = 0.5$ to give equal importance and scaling the score to [0,1]).

$$score(c) = \alpha \cdot Rank(c) + (1 - \alpha) \cdot \sum_{V_i = V_{start}}^{V_{end}} addCos(\mathbf{V_i}) + CS(\mathbf{N}(\mathbf{V_i})) \quad (5)$$

where, $addCos(\mathbf{V_i})$ is the *addCos* score of the vertex V_i and $N(V_i)$ is the neighbours of the vertex V_i from the word-graph G.

4 Multi-Document Abstractive Summarization

In this section, we apply our sentence level paraphrastic fusion model to generate multi-document level abstractive summary under a certain length limit (L). Our system takes a set of related documents as input and preprocesses them which includes tokenization, Part-Of-Speech (POS) tagging, removal of stopwords, filtering punctuation marks and Lemmatization. We use **NLTK** toolkit[3] to preprocess each sentence to obtain a more accurate representation of the information. In the following, we successively describe each of the steps involved in the document summarization process.

4.1 Sentence Clustering

The sentence clustering step allows us to group similar sentences. We use a hierarchical agglomerative clustering (Murtagh and Legendre, 2014) with a complete linkage criteria. This method proceeds incrementally, starting with each sentence considered as a cluster, and merging the pair of similar clusters after each step using bottom up approach. The complete linkage criteria determines the metric used for the merge strategy, which means largest distance between a sentence in one cluster and a sentence in the other candidate cluster. In building the clusters, we use the cosine similarity between the sentence embeddings obtained from equation (5). We set a similarity threshold ($\tau = 0.5$) to stop the clustering process by using a hold out dataset **SICK**[4] of SemEval-2014 (Marelli et al., 2014) for getting optimal performance. If we cannot find any cluster pair with a similarity above the threshold ($\tau = 0.5$), the process stops, and the clusters are released. The clusters may be small, but are highly coherent as each sentence they contain must be similar to every other sentence in the same cluster. This sentence clustering step is very important due to two main reasons,

- Selecting at most one sentence from each cluster of related sentences will decrease redundancy from the summary side.
- Selecting sentences from the diverse set of clusters will increase the information coverage from the document side as well.

[2]Available : http://www.keithv.com/software/giga/ (We used the 64K NVP vocabulary version)
[3]http://www.nltk.org/
[4]http://clic.cimec.unitn.it/composes/sick.html

For each cluster of related sentences, we generate 10-best (N = 10) abstractive fused sentences using our model described in section 3, the generated sentences differ in lengths as well as information. However, for the clusters containing only one sentence, we use our context sensitive lexical substitution model presented in section 3.3 to generate just the abstractive version of the source sentence.

4.2 Abstractive Sentence Selection

In our work, we use the concept-based ILP framework introduced in (Gillick and Favre, 2009) with some suitable changes to select the best subset of sentences. This approach aims to extract sentences that cover as many important concepts as possible, while ensuring the summary length is within a given budgeted constraint. Unlike (Gillick and Favre, 2009) which uses bigrams as concepts, we use keyphrases as concepts. Keyphrases are the words or phrases that represent the main topics of a document. Sentences containing the most relevant keyphrases are important for the summary generation. We extract the keyphrases from the document cluster using **RAKE**[5] (Rose et al., 2010). We assign a weight to each keyphrase using the score returned by RAKE.

Let $\bar{w}_i$ be the weight of keyphrase i and k_i a binary variable that indicates the presence of keyphrase i in the selected parafused sentences. Let l_j be the number of words or characters in sentence j, s_j a binary variable that indicates the presence of sentence j in the selected parafused sentence set and L the length limit for the set. Let Occ_{ij} indicate the occurrence of keyphrase i in sentence j, the ILP formulation is,

$$max : (\sum_i \bar{w}_i k_i + \sum_j (score(s_j) + \frac{l_j}{L}) \cdot s_j) \tag{6}$$

$$Subject\ to : \sum_j l_j s_j \leq L \tag{7}$$

$$s_j Occ_{ij} \leq k_i, \quad \forall i, j \tag{8}$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i \tag{9}$$

$$\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c \tag{10}$$

$$k_i \in \{0, 1\} \quad \forall i \tag{11}$$

$$s_j \in \{0, 1\} \quad \forall j \tag{12}$$

We try to maximize the weight of the keyphrases and our **ParaFuse** model's score (6), while avoiding repetition of those keyphrases (8, 9) and staying under the maximum number of words or characters allowed for the selected parafused sentences (7). In order to ensure at most one sentence per parafused cluster in the summary, we add an extra constraint (10), this will ensure non-redundancy from the summary side. In this process, we select the optimal combination of abstractive sentences that maximize information coverage while minimizing redundancy.

[5]https://github.com/aneesha/RAKE

4.3 Summary Length Limit Problem

One of the essential properties of the text summarization systems is the ability to generate a summary with a fixed length (**DUC 2004**, Task-2 (Multi-Document): Length limit = 100 Words). According to (Hong et al., 2014) all the multi-document summarizer from the previous research either truncated the summary to 100^{th} word, or removed the last sentence from the summary set. However, the first option produces a certain ungrammatical sentence, the later one can lose a lot of information in the worst case, if the sentences are long. Recently, (Kikuchi et al., 2016) propose four methods in order to tackle this issue, two of them are based on different decoding procedures without model architecture modification and the other two are learning-based, i.e., the models take the desired length information as input and encode it into the model architecture. However, their model is limited to headline generation task, where models generate a single sentence headline of a document. In this work, we tackle this issue in multi-document setting by generating N-best paraphrastic fusion length variations of a cluster of related sentences. Our model can effectively produce different length variations because of the shortest path strategy from start node to end node (see section 3.1 for the examples). In our ILP formulation for the document level summary generation, we try to maximize the total summary length in the objective function (equation (6)) to optimally solve the length limit problem. Under any circumstances, our model can choose a shorter variation of a sentence automatically to be included in the summary.

4.4 Experiments

In this section, we present our experimental details for assessing the performance of the sentence level paraphrastic fusion model and multi-document level abstractive summarization system as described above. We give details on the datasets we used, evaluation metrics, and the baseline systems used for comparison with our approach.

4.4.1 Sentence Level Experiments

We generate 50 shortest paths from start to end node for each cluster of related sentences using our paraphrastic sentence fusion model. The paths shorter than eight words or that do not contain a verb are filtered. To ensure pure abstractive compression generation, we remove the paths that have $cosineSimilarity \geq 0.9$ to any of the original sentence in the cluster. We then select 3-best candidates from K paths using the scoring function in equation (5). For fair evaluation, we also select the 3-best candidates for the baseline systems that we compare with our model.

Dataset: We conduct experiments on the human generated sentence fusion dataset released by (McKeown et al., 2010). This dataset consists of 300 English sentence pairs taken from newswire clusters accompanied by human-produced sentence fusions rewrites collected via Amazon's Mechanical Turk service[6]. We filtered the sentences which have no main verbs. The resulting set contains 296 pairs of sentences.

Evaluation Metric: We evaluate our system automatically using various automatic metrics. **BLEU** (Papineni et al., 2002) is the most commonly used metric for Machine Translation evaluation. BLEU relies on exact matching of n-grams and has no concept of synonymy or paraphrasing. **SARI** (Xu et al., 2016) a recently proposed metric which compares **S**ystem output **A**gainst **R**eferences and against the **I**nput sentence. SARI computes the arithmetic average of n-gram precision and recall of three rewrite operations: addition, copying, and deletion which correlates well with human references. **METEOR-E**[7] (Servan et al., 2016) is an augmented version of METEOR (Denkowski and Lavie, 2014) using distributed representations which can easily measure the abstractiveness. **Compression Ratio** is a measure of how terse a compression. A compression ratio of zero implies that the source sentence is fully uncompressed. We define **Copy Rate** as how many tokens are copied to the abstract sentence from the source sentence without paraphrasing in the following equation (13). Lower copy rate score means more paraphrasing is involved in the abstract sentence. Copy rate of 100% means no paraphrasing.

[6] http://www.mturk.com

[7] https://github.com/cservan/METEOR-E

Input Sentences	Bush, who initially nominated Roberts to replace retiring Justice Sandra Day O'Connor, tapped him to lead the court the day after Rehnquist's death. President Bush initially nominated Roberts in July to succeed retiring Justice Sandra Day O'Connor.
(Filippova, 2010)	president bush initially nominated roberts to replace retiring justice sandra day o'connor .
(Boudin and Morin, 2013)	bush , who initially nominated roberts in july to succeed retiring justice sandra day o'connor , tapped him to lead the court the day after rehnquist 's death .
(Banerjee et al., 2015)	bush , who initially nominated roberts to replace retiring justice sandra day o'connor , tapped him to lead the court the day after rehnquist 's death .
Paraphrastic Fusion (*ours*)	president bush initially **recommended** roberts in july to **substitute** retiring justice sandra day o'connor , tapped him to **run** the court the day after rehnquist 's death .

Table 1: The output generated by the baseline and our system (the paraphrased words are marked bold)

Model	BLEU	SARI	METEOR-E	Compression Ratio	Copy Rate
(Filippova, 2010)	40.6	34.6	0.31	**0.57**	99.8
(Boudin and Morin, 2013)	**44.0**	37.2	0.36	0.42	99.9
(Banerjee et al., 2015)	42.3	36.5	0.34	0.45	99.8
Paraphrastic Fusion (*ours*)	42.5	**37.4**	**0.43**	0.41	**76.2**

Table 2: Comparison with baselines and our **Paraphrastic Fusion** model across different automatic evaluation metrics (the scores are averaged)

$$Copy\ Rate = \frac{|S_{orig} \cap S_{abs}|}{|S_{abs}|} \tag{13}$$

4.5 Baseline Systems and Results

We compare our system with (Filippova, 2010), (Boudin and Morin, 2013)[8] and (Banerjee et al., 2015)[9]. Table 1 shows the output generated by the baseline and our system. We report our system's performance compared with the baselines in terms of different evaluation metrics in Table 2. Our model balances the information coverage (**BLUE**, **SARI**) and complete abstractiveness (**METEOR-E**, **Copy Rate**) instead of over compressing the generated sentences (**Compression Ratio**). We get slightly higher score in **SARI** because of the multiple human abstractive rewrites along with input sentence. The **Copy Rate** score of other baseline systems clearly indicates the fact that they are doing completely deletion based compression, no paraphrasing is involved. Moreover, we also get higher score in **METEOR-E** metric because of the lexical substitution operation. As expected, we get little lower **BLEU** score compared to (Boudin and Morin, 2013) for two main reasons (1) We tried to balance between information coverage and abstractiveness (2) **BLEU** works well on surface level lexical overlap.

4.5.1 Multi-Document Level Experiments

Dataset: We consider the generic multi-document summarization dataset provided at Document Understanding Conference (**DUC 2004**)[10] which is one of the main benchmark dataset in the multi-document summarization field. It contains 50 document clusters and each is composed of 10 news wire articles about a given topic from the Associated Press and The New York Times that are published between 1998 to 2000. The dataset also contains multiple human-written summaries which are used for the evaluation of system-generated summaries. The **Opinosis** (Ganesan et al., 2010) is another dataset consists of short user reviews in 51 different topics collected from TripAdvisor, Amazon, and Edmunds. The dataset is well suited for multi-document summarization which includes 5 different golden summaries for each topic created by human authors.

[8]https://github.com/boudinfl/takahe

[9]https://github.com/StevenLOL/AbTextSumm

[10]http://duc.nist.gov/duc2004/

Dataset	Models	R-1	R-2	R-WE-1	R-WE-2
DUC 2004	LexRank (Erkan and Radev, 2004)	35.95	7.47	36.91	7.91
	Submodular (Lin and Bilmes, 2011)	39.18	9.35	40.03	9.92
	RegSum (Hong and Nenkova, 2014)	38.57	9.75	39.12	10.33
	ILPSumm (Banerjee et al., 2015)	39.24	11.99	40.21	12.08
	PDG* (Yasunaga et al., 2017)	38.45	9.48	39.07	10.24
	ParaFuse_doc (*ours*)	**40.07**	**12.04**	**42.31**	**12.96**
Opinosis 1.0	TextRank (Mihalcea and Tarau, 2004)	27.56	6.12	28.20	6.45
	Opinosis (Ganesan et al., 2010)	32.35	9.13	33.54	9.41
	Biclique (Muhammad et al., 2016)	33.03	8.96	33.91	9.25
	ParaFuse_doc (*ours*)	**33.86**	**9.74**	**34.46**	**10.09**

Table 3: Results on DUC 2004 (Task-2) and Opinosis 1.0

Evaluation Metric: We evaluate our summarization system using **ROUGE**[11] (Lin, 2004) on **DUC 2004** (Task-2, Length limit (L) = 100 Words) and **Opinosis 1.0** (L = 15 Words). However, ROUGE scores are unfairly biased towards lexical overlap at surface level. Taking this into account, we also evaluate our system using **ROUGE-WE** (Ng and Abrecht, 2015), which considers word embeddings to compute the semantic similarity of the words. We report limited length recall performance for both the metrics, as our system generated summaries are forced to be concise through some constraints (such as length limit constraint). Therefore, we consider using just the recall score since precision is of less concern in this scenario.

4.5.2 Baseline Systems & Results

The summaries generated by the baseline **LexRank** (Erkan and Radev, 2004) and the state-of-the-art summarizers (**Submodular** (Lin and Bilmes, 2011) and **RegSum** (Hong and Nenkova, 2014)) on the DUC 2004 dataset were collected from (Hong et al., 2014). In case of **ILPSumm**[12] (Banerjee et al., 2015) and **PDG*** (Yasunaga et al., 2017), we use the author provided implementation to generate summary from their model. For Opinosis 1.0 dataset, we use an open source implementation of **TextRank** (Mihalcea and Tarau, 2004)[13]. Moreover, we use the author provided implementation for the **Opinosis** (Ganesan et al., 2010) and **Biclique** (Muhammad et al., 2016) to generate summaries. According to the Table 3, our multi-document level model **ParaFuse_doc** achieves the best summarization performance on all the ROUGE metrics for both the datasets. The slight increase in terms of **R-WE** metric clearly justifies the fact of abstractiveness proposed in this work which highly correlates with human references.

5 Conclusion

In this paper, we designed a new abstractive fusion generation model at the sentence level which jointly performs sentence fusion and paraphrasing. Our sentence level model is very well suited for full abstractive multi-document summarization which was justified by the experimental results on two benchmark datasets of different domains. Furthermore, we designed an optimal solution for the classical summary length limit problem in multi-document setting.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments. The research reported in this paper was conducted at the University of Lethbridge and supported by the Natural Sciences and Engineering Research Council (**NSERC**) of Canada discovery grant and the University of Lethbridge.

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

[12] https://github.com/StevenLOL/AbTextSumm

[13] https://github.com/davidadamojr/TextRank

References

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

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1208–1214. AAAI Press.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Comput. Linguist.*, 31(3):297–328, September.

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 298–305, Atlanta, Georgia, June. Association for Computational Linguistics.

Yllias Chali, Moin Tanvee, and Mir Tafseer Nayeem. 2017. Towards abstractive multi-document summarization using submodular function-based framework, sentence compression and merging. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*, pages 418–424.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. pages 103–111, October.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June. Association for Computational Linguistics.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 377–384, Stroudsburg, PA, USA. Association for Computational Linguistics.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

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

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 177–185, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 340–348, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1693–1701, Cambridge, MA, USA. MIT Press.

Kai Hong and Ani Nenkova. 2014. 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, Gothenburg, Sweden, April. Association for Computational Linguistics.

Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1070.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas, November. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 2017a. Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2090, Copenhagen, Denmark, September. Association for Computational Linguistics.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017b. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark, September. Association for Computational Linguistics.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Shuming Ma, Xu Sun, Jingjing Xu, Houfeng Wang, Wenjie Li, and Qi Su. 2017. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 635–640, Vancouver, Canada, July. Association for Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–320, Los Angeles, California, June. Association for Computational Linguistics.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, Colorado, June. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Azam Sheikh Muhammad, Peter Damaschke, and Olof Mogren. 2016. Summarizing online user reviews using bicliques. In *Proceedings of the 42Nd International Conference on SOFSEM 2016: Theory and Practice of Computer Science - Volume 9587*, pages 569–579, New York, NY, USA. Springer-Verlag New York, Inc.

Fionn Murtagh and Pierre Legendre. 2014. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *J. Classif.*, 31(3):274–295, October.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, page 280.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*.

Shashi Narayan, Nikos Papasarantopoulos, Shay B. Cohen, and Mirella Lapata. 2018b. Neural extractive summarization with side information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, February.

Mir Tafseer Nayeem and Yllias Chali. 2017a. Extract with order for coherent multi-document summarization. In *Proceedings of TextGraphs@ACL 2017: the 11th Workshop on Graph-based Methods for Natural Language Processing, Vancouver, Canada, August 3, 2017*, pages 51–56.

Mir Tafseer Nayeem and Yllias Chali. 2017b. Paraphrastic fusion for abstractive multi-sentence compression generation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 2223–2226.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal, September. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In Michael W. Berry and Jacob Kogan, editors, *Text Mining. Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July. Association for Computational Linguistics.

Christophe Servan, Alexandre Berard, zied elloumi, Hervé Blanchon, and Laurent Besacier. 2016. Word2vec vs dbnary: Augmenting meteor using vector representations or lexical resources? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1159–1168, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond K. Wong, and Fang Chen. 2016. An efficient approach for multi-sentence compression. In *Proceedings of The 8th Asian Conference on Machine Learning*, volume 63 of *Proceedings of Machine Learning Research*, pages 414–429, The University of Waikato, Hamilton, New Zealand, 16–18 Nov. PMLR.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 291–297, Valencia, Spain, April. Association for Computational Linguistics.

Dung Tran Tuan, Nam Van Chi, and Minh-Quoc Nghiem, 2017. *Multi-sentence Compression Using Word Graph and Integer Linear Programming*, pages 367–377. Springer International Publishing, Cham.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada, August. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada, July. Association for Computational Linguistics.