Abstractive Unsupervised Multi-Document Summarization using Paraphrastic Sentence Fusion

Mir Tafseer Nayeem Tanvir Ahmed Fuad Yllias Chali
University of Lethbridge, Alberta, Canada

(COLING 2018)

Outline

- Extractive vs Abstractive Summarization
- Single Document vs Multi-Document Summarization
- Challenges
- Methodology
  - Multi-Sentence Compression
  - Lexical Paraphrasing
  - ILP
- Experiment
- Final Remark

Extractive vs Abstractive Summarization

Extractive Approach:

Identify the most important sentences in the document and copy that directly into the final summary.

Abstractive Approach:

Understand the most important content in the document. Write it in your own way.

- Involves Natural Language Generation Techniques
- Closer to human-like interpretation

Single Document vs Multi Document Summarization

Differences between SDS and MDS:

- More diverse input document types
- Insufficient methods to capture cross-document relations
- High redundancy and contradiction across input documents
- Larger searching space but lack of sufficient training data
- Lack of evaluation metrics specifically designed for MDS

Challenges Addressed

The information overlap between the documents from the same topic

- the extractive methods would produce redundant summary or biased towards specific source document

Multi-Sentence Compression and Lexical Paraphrasing

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).

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

Word Graph Construction for Sentence Fusion

Given a cluster 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.

Word Graph

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.

Possible 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%.

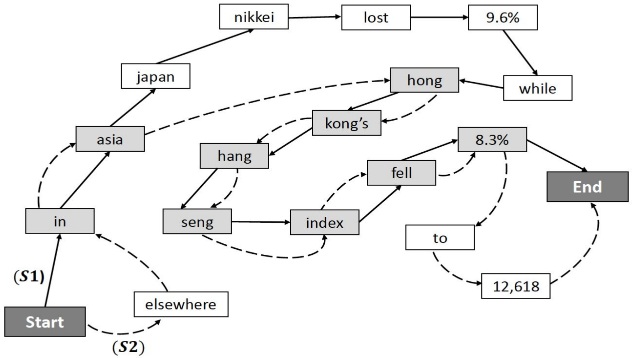


Figure 1: Constructed Word graph and a possible compression path (light gray nodes)

Challenges in Word Graph

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.

- We need a candidate ranking strategy to sort the generated K-shortest paths based on the information coverage.

Candidate Ranking

Candidates are ranked using the **Textrank** algorithm.

How does Textrank work?

1. A graph is constructed where the vertices of the graph represent each sentence in a document and the edges between sentences are based on content overlap, namely by calculating the number of words that 2 sentences have in common.
2. Based on this network of sentences, the sentences are fed into the Pagerank algorithm which identifies the most important sentences.

Textrank returns an importance score for each sentence.

TextRank Limitation

Lexical Overlap is not guaranteed even when two sentences are semantically similar.

Solution

Use cosine similarity of sentence embeddings as edge weight.

How to get sentence embeddings:

Word2vec 300d embeddings for every word in a sentence. Encode the sentence using GRU, concatenate the forward and backward hidden vector for the last token.

Context Sensitive Lexical Substitution

Target Word Identification for Substitution

Nouns and Verbs for possible substitution candidates from the word-graph G. Named entities of type PER, LOC, ORG, MISC are not considered for the substitution.

Substitution Selection:

- Gather lexical substitution set S for a target word from PPDB.
- Hardcode the model to select substitutes with the same POS tag and that are not a morphological variant (e.g., fly, flew, flown).

Context Sensitive Lexical Substitution

Substitution Ranking

The authors use Word and context vectors released by (Melamud et al., 2015) which was shown to perform strongly on lexical substitution task.

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 = (c1; c2; ....; cn), which is defined as follows,

$$\text{addCos}(s \mid t, C) = (\cos(s,t) + \textstyle\sum_{c \subset C} \cos(s,c)) \,/\, |C| + 1$$

best substitution s according to maximum addCos scores over 0.7 is attached
with the target word vertex t in the word-graph G along with the addCos score.

Context Sensitive Lexical Substitution

Confidence Score:

For each substitute word, confidence score is calculated using a Language model.

- The authors use a 3-gram LM, meaning, each substitute target is accompanied by the adjacent vertices (words)

Finally, we rank the K candidate fusions and find the N-best paraphrastic sentence fusion which balances the information coverage and the abstractiveness.

$$\text{score}(c) = \boldsymbol{\alpha}\ \text{Rank}(c) + (1-\boldsymbol{\alpha}) \quad \sum^{V\_end}_{V\_start} \text{addCos}(Vi) + \text{CS}(N(Vi))$$

Multi Document Abstractive Summarization (MDAS)

Apply the sentence level paraphrastic fusion model to generate multi-document level abstractive summary under a certain length limit (L).

Preprocessing:

The 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 using NLTK toolkit.

MDAS (Sentence Clustering)

Hierarchical Agglomerative Clustering approach

- starting with each sentence considered as a cluster, and merging the pair of similar clusters after each step using bottom up approach.
- In building the clusters, cosine similarity is used between the sentence embeddings.
- A similarity threshold (= 0:5) to stop the clustering process by using a hold out dataset SICK4 of SemEval-2014 (Marelli et al., 2014) for getting optimal performance.
- If no cluster pair with a similarity above the threshold (= 0:5) exists, the process stops, and the clusters are released.

MDAS (Sentence Clustering - Importance)

- 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.

For each cluster of related sentences, 10-best (N = 10) abstractive fused sentences using the sentence fusion model are generated.

MDAS (ILP)

Aims to extract sentences that cover as many important concepts as possible, while ensuring the summary length is within a given budgeted constraint.

Keyphrases are the words or phrases that represent the main topics of a document. Sentences containing the most relevant key phrases are important for the summary generation.

- Extract the keyphrases from the document cluster using RAKE and assign a weight to each keyphrase using the score returned by RAKE.

MDAS (ILP)

- w_i be the weight of keyphrase i
- k_i a binary variable that indicates the presence of keyphrase i in the selected parafused sentences.
- l_j be the number of words or characters in sentence j
- sj a binary variable that indicates the presence of sentence j in the selected parafused sentence set.
- L is the length limit for the set.
- Let Occ_{ij} indicate the occurrence of keyphrase i in sentence j

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

$$Subject\ to : \sum_j l_j s_j \leq L$$

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

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

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

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

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

Experiments - Sentence Level

Settings:

- 50 shortest paths from start to end node for each cluster of related sentences using the sentence fusion model. The paths shorter than eight words or that do not contain a verb are filtered.

- To ensure pure abstractive compression generation, remove the paths that have cosine similarity of 0.9 to any of the original sentence in the cluster.

- Select 3-best candidates from K paths using the scoring function

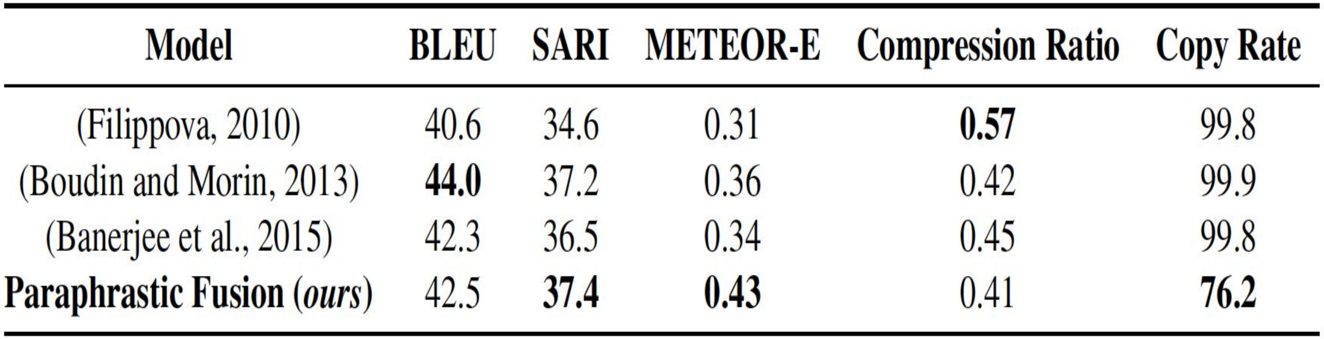
Experiments - Sentence Level (Dataset)

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.

Experiments - Sentence Level (Evaluation Metric)

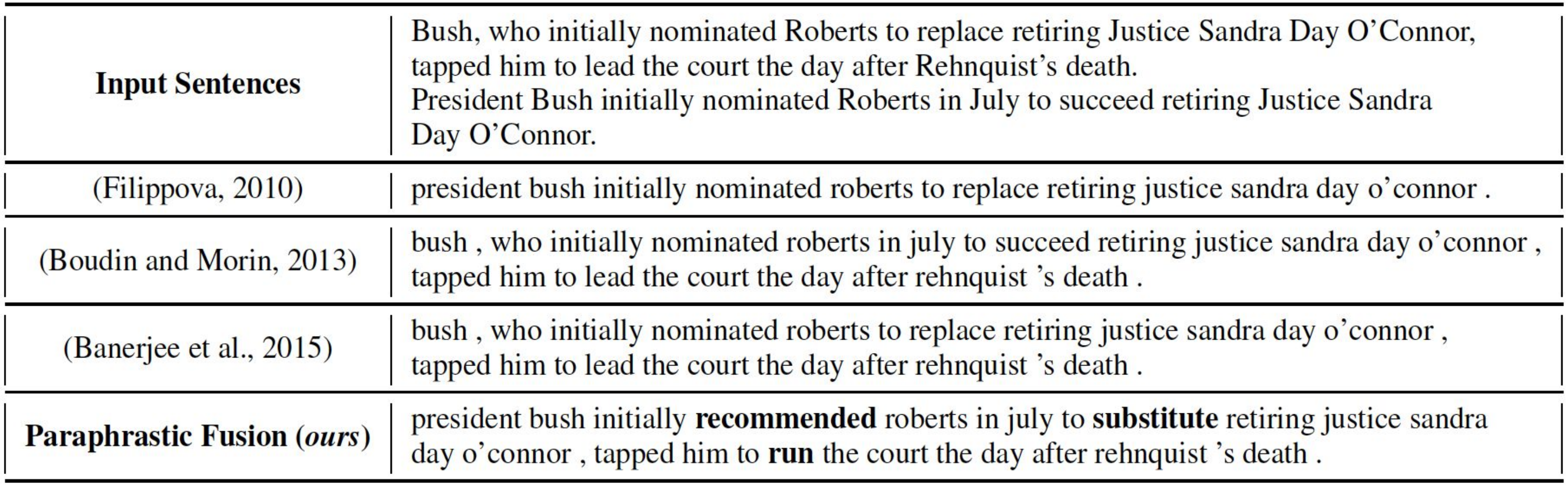
1. BLEU : relies on exact matching of n-grams and has no concept of synonymy or paraphrasing.
2. SARI : computes the arithmetic average of n-gram precision and recall of three rewrite operations: addition, copying, and deletion which correlates well with human references.
3. METEOR-E: using distributed representations which can easily measure the abstractiveness.
4. Compression Ratio : a measure of how terse a compression.
5. Copy Rate : how many tokens are copied to the abstract sentence from the source sentence without paraphrasing in the following equation

Experiments - Sentence Level(Comparison against Baselines)



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**

Experiments - Sentence Level(Example)



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)

Experiments - MDAS (Dataset)

The generic multi-document summarization dataset provided at Document Understanding Conference (DUC 2004)

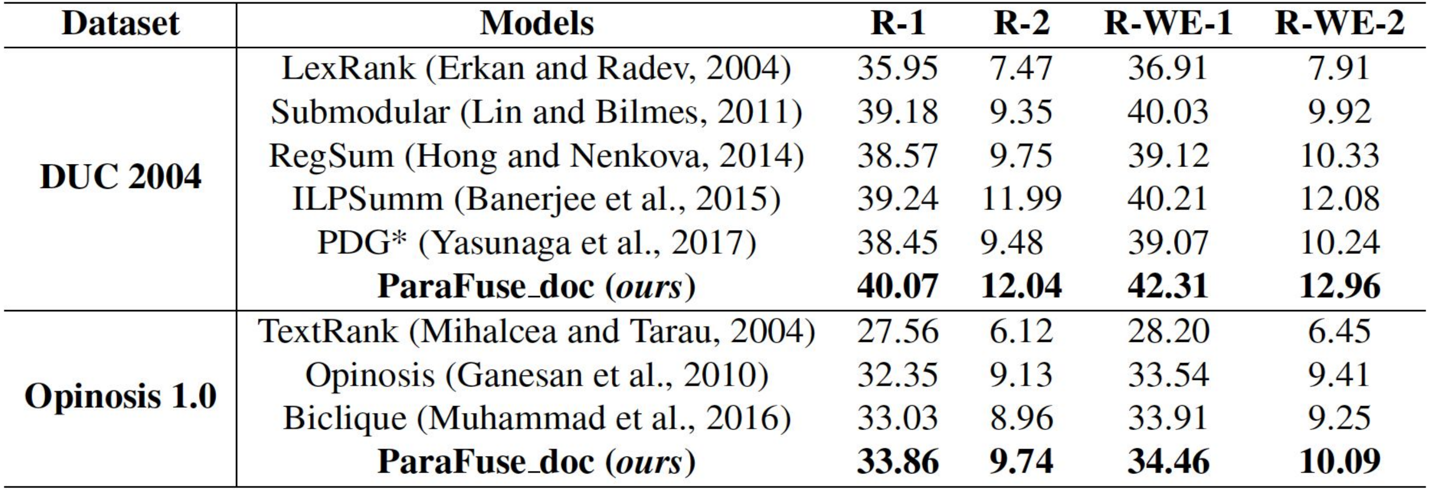
- Contains 50 document clusters and each is composed of 10 news wire articles about a given topic from the Associated Press and The New York Times that are published between 1998 to 2000.
- Contains multiple human-written summaries which are used for the evaluation of system-generated summaries.

The Opinosis (Ganesan et al., 2010)

- Consists of short user reviews in 51 different topics collected from TripAdvisor, Amazon, and Edmunds.
- Includes 5 different golden summaries for each topic created by human authors.

Experiments - MDAS

ROUGE-WE (Ng and Abrecht, 2015) considers word embeddings to compute the semantic similarity of the words.



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

Thank You