

Multi-Document Abstractive Summarization Using ILP based Multi-Sentence Compression

Siddhartha Banerjee

The Pennsylvania State University
University Park, PA, USA
sub253@ist.psu.edu

Prasenjit Mitra

QCRI
Doha, Qatar
pmitra@qf.org.qa

Kazunari Sugiyama

National University of Singapore
Singapore
sugiyama@comp.nus.edu.sg

Abstract

Abstractive summarization is an ideal form of summarization since it can synthesize information from multiple documents to create concise informative summaries. In this work, we aim at developing an abstractive summarizer. First, our proposed approach identifies the most important document in the multi-document set. The sentences in the most important document are aligned to sentences in other documents to generate clusters of similar sentences. Second, we generate K -shortest paths from the sentences in each cluster using a word-graph structure. Finally, we select sentences from the set of shortest paths generated from all the clusters employing a novel integer linear programming (ILP) model with the objective of maximizing information content and readability of the final summary. Our ILP model represents the shortest paths as binary variables and considers the length of the path, information score and linguistic quality score in the objective function. Experimental results on the DUC 2004 and 2005 multi-document summarization datasets show that our proposed approach outperforms all the baselines and state-of-the-art extractive summarizers as measured by the ROUGE scores. Our method also outperforms a recent abstractive summarization technique. In manual evaluation, our approach also achieves promising results on informativeness and readability.

1 Introduction

Abstractive summarization has gained popularity due to its ability of generating new sentences to convey the important information from text documents. An abstractive summarizer should present the summarized information in a coherent form that is easily readable and grammatically correct. Readability or linguistic quality is an important indicator of the quality of a summary. Several text-to-text (T2T) generation techniques that aim to generate novel text from textual input have been developed [Knight and Marcu, 2002; Zajic *et al.*, 2007; Barzilay and McKeown, 2005]. However, to the best of our knowledge, none of the above methods explicitly model the role of linguistic quality and only aim at

maximizing information content of the summaries. In this work, we address readability by assigning a log probability score from a language model as an indicator of linguistic quality. More specifically, we build a novel optimization model for summarization that jointly maximizes information content and readability.

Extractive summarizers [Mani and Maybury, 1999] often lose a lot of information from the input as they only “extract” a few important sentences from the documents to create the final summary. We prevent information loss by aggregating information from multiple sentences. We generate clusters of similar sentences from a collection of documents. Multi-sentence compression (MSC) [Filippova, 2010] can be used to fuse information from sentences in a cluster. However, MSC might generate sentences that convey similar information from two different clusters. By contrast, our Integer Linear Programming (ILP) based approach prevents redundant information from being included in the summary using an inter-sentence redundancy constraint. Consequently, our experiments reveal that our method generates more informative and readable summaries than MSC.

Our proposed approach to abstractive summarization consists of the following two steps: (1) Aligning similar sentences from multiple-documents and (2) Generating the most informative and linguistically well-formed sentence from each cluster, and then appending them together. In multi-document summarization, all documents are not equally important; some documents contain more information on the main topics in the document set. Our first step estimates the importance of a document in the whole dataset using *LexRank* [Erkan and Radev, 2004], *Pairwise Cosine Similarity* and *Overall Document Collection Similarity*. Each sentence from the most important document are initialized into separate clusters. Thereafter, each sentence from the other documents are assigned to the cluster that has the highest similarity with the sentence. In the generation step, we first generate a word-graph structure from the sentences in each cluster and construct K shortest paths from the graph between the start and end nodes. We formulate a novel integer linear programming (ILP) problem that maximizes the information content and linguistic quality of the generated summary. Our ILP problem represents each of the K shortest paths as a binary variable. The coefficients of each variable in the objective function is obtained by combining the information score

of the path and the linguistic quality score. We introduce several constraints into our ILP model. We ensure that only one sentence is generated from each cluster. Second, we avoid redundant sentences that carry the same or similar information from different clusters. The solution to the optimization problem decides the paths that would be included in the final abstractive summary.

On the DUC2004 and DUC2005 datasets, we demonstrate the effectiveness of our proposed method. Our proposed method outperforms not only some popular baselines but also the state-of-the-art extractive summarization systems. ROUGE scores [Lin, 2004] obtained by our system outperforms the best extractive summarizer on both the datasets. Our method also outperforms an abstractive summarizer based on multi-sentence compression [Filippova, 2010] when measured by ROUGE-2, ROUGE-L and ROUGE-SU4 scores. Further, manual evaluation by human judges shows that our technique produces summaries with acceptable linguistic quality and high informativeness.

2 Related Work

Several researchers have developed abstractive summarizers. Genest and Lapalme [2010] used natural-language-generation (NLG) systems. However, NLG requires a lot of manual effort in terms of defining schemas as well as using deeper natural language analysis. Wang and Cardie [2013] and Oya *et al.* [2014] induced templates from the training set in their meeting summarization tasks. Such induction of templates, however, is not very effective in news summarization because of the variability in topics. Unlike these methods, our method does not induce any templates but generates summaries in an unsupervised manner by combining information from several sentences on the same topic. Berg-Kirkpatrick *et al.* [2011] used an ILP formulation that jointly extracts and compresses sentences to generate summaries. However, their method is supervised and requires significant manual effort to define features for subtree deletions, which is required to compress sentences. Graph-based techniques have also been very popular in summarization. Ganesan *et al.* [2010] employed a graph-based approach to generate concise abstractive summaries from highly redundant opinions. Compared with their opinionated texts such as product reviews, the target documents in multi-document summarization do not contain such high level of redundancy.

More recently, Mehdad *et al.* [2013] proposed a supervised approach for meeting summarization, in which they generate an entailment graph of sentences. The nodes in the graph are the linked sentences and edges are the entailment relations between nodes; such relations help to identify non-redundant and informative sentences. Their fusion approach used MSC [Filippova, 2010], which generates an informative sentence by combining several sentences in a word-graph structure. However, Filippova’s method produces low linguistic quality as the ranking of generated sentences is based on edge weights calculated only using word collocations. By contrast, our method selects sentences by jointly maximizing informativeness and readability and generates informative, well-formed and readable summaries.

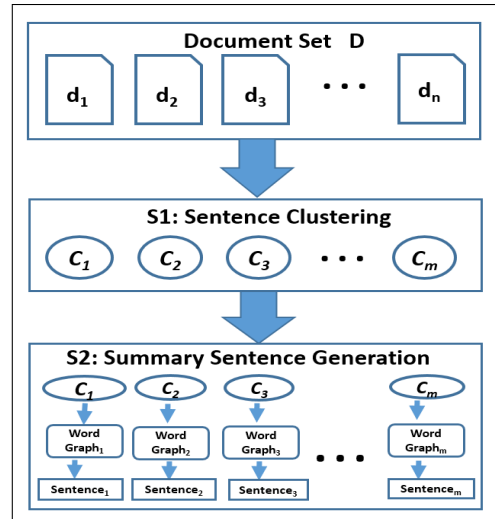


Figure 1: Overview of our abstractive summarization approach.

3 Proposed Approach

Figure 1 shows our proposed abstractive summarization approach, which consists of the following two steps:

S1: *Sentence clustering*,

S2: *Summary sentence generation*.

Given a document set D , that consists of n documents ($d_1, d_2, d_3, \dots, d_n$), our approach first generates m clusters ($C_1, C_2, C_3, \dots, C_m$) of similar sentences, and then use the individual clusters to create word-graphs. A maximum of one novel sentence is generated from each word-graph with the goal of maximizing information content and linguistic quality of the entire summary. The sentence clustering step (S1) has two important components: the first (S1-1) identifies the most important document D_{imp} in D before the final cluster generation step (S1-2) that generates clusters of similar sentences. We experiment with several techniques to identify D_{imp} , and then align sentences from other documents to the sentences in D_{imp} . It proves to be a simple, yet effective technique for generating clusters containing similar information. Our approach is inspired by the findings of Wan [2008] that showed how the incorporation of document impact can improve the performance of summarization. In (S2), we create a directed word-graph structure from the sentences in each cluster. In the word-graph, the nodes represent the words and the edges define the adjacency relations in the sentences. From the word-graph, multiple paths between the start and the end nodes can be extracted. We extract K shortest paths from each cluster, and finally retain the paths that maximize information content and linguistic quality using an ILP based approach. We impose constraints on the maximum number of sentences that are generated from each cluster and also impose constraints to avoid redundancies such that similar information from different clusters are not included in the summary. Information content is measured using TextRank [Mihalcea and Tarau, 2004], which scores sentences based on the presence of keywords. We measure linguistic quality using a 3-gram language model that assigns confidence values to

sequences of words in the sentences. In this section, we describe both steps – S1 and S2.

S1: Sentence Clustering

We initialize clusters of sentences using each sentence from the most important document, D_{imp} , in a document set D . Our intuition behind this approach is that D_{imp} consists of the most important content relevant across all the documents in D . In other words, the document that is most close to the central content of the collection is the most informative.

(S1-1) Document Importance

We propose several techniques to identify D_{imp} .

LexRank ($MD_{LexRank}^{Imp}$): LexRank [Erkan and Radev, 2004] constructs a graph of sentences where the edge weights are obtained by the inter-sentence cosine similarities. While the original LexRank constructs a graph of sentences, we construct a graph of documents to compute document importance. Equation (1) shows how LexRank scores are computed using weighted links between the nodes (documents). This equation measures the salience of a node in the graph, which is the importance of the document in the entire document collection. Let $p(u)$ be the centrality of node u . LexRank is then defined as follows:

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in adj[u]} \frac{\text{idf-modified-cosine}(u,v)}{\sum_{z \in adj[v]} \text{idf-modified-cosine}(z,v)} p(v), \quad (1)$$

where $adj[u]$ and N are the set of nodes that are adjacent to u and the total number of nodes in the graph, respectively. The damping factor d is usually set to 0.85, and we set d to this value in our implementation. D_{imp} is determined as the document that has the highest LexRank score in D once the above equation converges.

Pairwise Cosine Similarity (MD_{CosSim}^{Imp}): This method computes the average cosine similarity between the target document d_i and the other documents in the dataset. The average similarity is calculated using the following formula:

$$AveCosSim(d_i) = \frac{\sum_{d_j \in D} CosSim(d_i, d_j)}{|D| - 1} \quad (i \neq j),$$

where $|D|$ denotes the number of documents in the document set D .

Overall document collection similarity ($MD_{DocSetSim}^{Imp}$): This method computes the cosine similarity between the target document and the whole document set. We create the whole document set by concatenating text from all the documents in D . This method is defined as follows:

$$DocSetSim(d_i) = CosSim(d_i, D).$$

In $MD_{LexRank}^{Imp}$, MD_{CosSim}^{Imp} , and $MD_{DocSetSim}^{Imp}$ mentioned above, we select the document D_{imp} with the highest score as the most important one in the dataset D . Next, we generate the clusters by aligning sentences and re-ordering them based on original positions of the sentences in the documents.

(S1-2) Cluster Generation

The sentences from each of the other documents ($d_i \neq D_{imp}$) in D are assigned to the clusters one-by-one based on cosine

similarity measure. Our approach computes pairwise cosine similarity of each sentence in d_i to all the sentences in D_{imp} . For example, a sentence in d_i , $s_p^{d_i}$ has the highest similarity with $s_j^{D_{imp}}$, a sentence in D_{imp} . Then, we assign $s_p^{d_i}$ to cluster C_j , in which $s_j^{D_{imp}}$ was initially assigned. Some sentences in d_i might not be similar to any of the sentences in D_{imp} . Hence, we only align sentences when the similarity > 0.5 . Further, we only retain clusters that have at least $|D|/2$ sentences, assuming that a content is relevant only if it exists in half of the documents in D .

Cluster ordering: We implement two *cluster ordering* techniques that reorder clusters based on the original position of the sentences in the documents.

1. **Majority ordering (MO):** Given two clusters, C_i and C_j , the set of common documents from which the sentences are assigned to the two clusters are identified. If C_i and C_j have sentences $s_p^{d_i}$ and $s_q^{d_i}$ ($p < q$), respectively, where d_i is the common document, then C_i precedes C_j . The final order is determined based on overall precedence of the sentences of one cluster over the others.
2. **Average position ordering (APO):** The sentences $\{s_p, s_q, \dots, s_z\}$ in any cluster C_i are each assigned a normalized score. For example, the normalized score of s_p is computed as the ratio of the original position of the sentence and the total number of sentences in d_i (here, s_p belongs to document d_i). When ordering two clusters, the cluster that has the lower score obtained by averaging the normalized scores of all the sentences is ranked higher than the others.

S2: Summary Sentence Generation

In order to generate a one-sentence representation from a cluster of redundant sentences, we use multi-sentence compression. We generate multiple sentences from a cluster using a word-graph [Filippova, 2010]. Suppose that a cluster C_i contains j sentences, $S = \{s_1, s_2, \dots, s_j\}$. A directed graph is created by adding sentences from S to the graph in an iterative fashion. Each sentence is connected to dummy *start* and *end* nodes to mark the beginning and ending of the sentences. The vertices or nodes are the words along with the parts-of-speech (POS) tags. We connect adjacent words in the sentences with directed edges. Once the first sentence is added, words from the following sentences are mapped onto a node in the graph provided that they have exactly the same word form and the same POS tag. The sequence of rules used for the word-graph construction is as follows:

- Content words are added for which there are no candidates in the existing graph,
- Content words for which multiple mappings are possible or such words that occur more than once in the sentence,
- Stopwords.

The context of the words are taken into consideration if multiple mappings are possible, and the word is mapped to that node that has the highest directed context. We also add punctuations to the graph. Figure 2 shows a simple example of the word-graph generation technique. We do not show POS and punctuations in the figure for clarity. Consider the

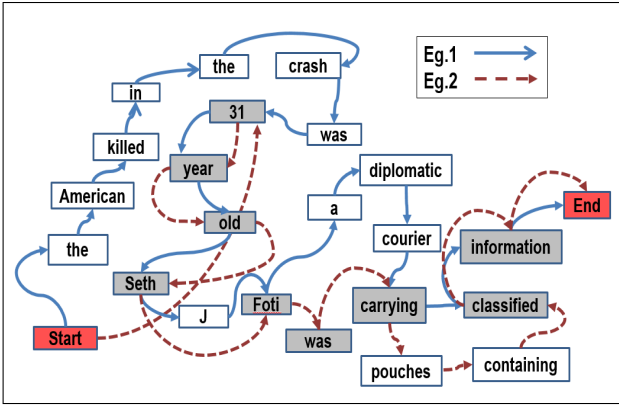


Figure 2: Word-graph generation from sentences.

following two sentences as an illustration of our generation approach:

Eg.1 *The American killed in the crash was 31-year-old Seth J. Foti, a diplomatic courier carrying classified information.*

Eg.2 *31-year-old Seth Foti was carrying pouches containing classified information.*

As shown in the examples above, the two sentences contain similar information, but they are syntactically different. The solid directed arrows connect the nodes in Eg.1, whereas the dotted arrows join the nodes in Eg.2. We can obtain several shortest paths between the start and end nodes. In Figure 2, we highlight one such path using gray rectangles. Several other paths are possible, for example:

- 1: *31 year old Seth Foti was carrying pouches containing classified information.*
- 2: *31 year old Seth Foti a diplomatic courier carrying classified information.*

The original input sentences from the cluster are also valid paths between the *start* and *end* nodes. To ensure pure abstractive summarization, we remove such paths that are same or very similar (cosine similarity ≥ 0.8) to any of the original sentences in the cluster. Similar to Filippova’s word-graph construction, we set the minimum path length (in words) to eight to avoid incomplete sentences. Finally, we retain a maximum of 200 randomly selected paths from each cluster to reduce computational overload of the ILP based approach. Our aim is to select the best path from all available paths.

Best Path Selection

From 200 paths in each cluster, we choose at most one path that maximizes information content and linguistic quality together. Let $p_i^{C_j}$ be each path in a cluster C_j , namely,

$$\forall j \in \{1, \dots, m\}, C_j \ni \{p_1^{C_j}, p_2^{C_j}, \dots, p_K^{C_j}\},$$

where the total number of shortest paths is equal to $K = \min[|C_j|, 200]$ where $|C_j|$ refers to the maximum number of paths that can be generated from a cluster. We argue that the shortest paths that we select in the final summary should be informative as well as linguistically readable. Hence, we introduce two factors – *Informativeness* ($I(p_i^{C_j})$) and *Linguistic quality* ($LQ(p_i^{C_j})$).

Informativeness: In principle, we can use any existing method that computes the importance of a sentence to define *Informativeness*. In our model, we use TextRank scores [Mihalcea and Tarau, 2004] to generate an importance value of a sentence within a cluster. TextRank creates a graph of words from the sentences. The score of each node in the graph is calculated as shown in Equation (2):

$$S(V_i) = (1 - d) + d \times \sum_{V_j \in \text{adj}(V_i)} \frac{w_{ji}}{\sum_{V_k \in \text{adj}(V_i)} w_{jk}} S(V_i), \quad (2)$$

where V_i represents the words, $\text{adj}(V_i)$ denotes the adjacent nodes of V_i and d is the damping factor set to 0.85. The computation converges to return final word importance scores. The informativeness score of a path ($I(p_i^{C_j})$) is obtained by adding the importance scores of the individual words in the path.

Linguistic Quality: In order to compute *Linguistic quality*, we use a language model. More specifically, we use a 3-gram (trigram) language model that assigns probabilities to sequence of words. Suppose that a path contains a sequence of q words $\{w_1, w_2, \dots, w_q\}$. The score $LQ(p_i^{C_j})$ assigned to each path is defined as follows:

$$LQ(p_i^{C_j}) = \frac{1}{1 - LL(w_1, w_2, \dots, w_q)}, \quad (3)$$

where $LL(w_1, w_2, \dots, w_q)$ is defined as:

$$LL(w_1, w_2, \dots, w_q) = \frac{1}{L} \cdot \log_2 \prod_{t=3}^q P(w_t | w_{t-1} w_{t-2}). \quad (4)$$

As can be seen from Equation (4), we obtain the conditional probability of different sets of 3-grams in the sentence. The scores are combined and averaged by L , the number of conditional probabilities computed. The $LL(w_1, w_2, \dots, w_q)$ scores are negative; with higher magnitude implying lower readability. Therefore, in Equation (3), we take the reciprocal of the logarithmic value with smoothing to compute $LQ(p_i^{C_j})$. In our experiments, we used a 3-gram model that is trained on the English Gigaword corpus¹.

ILP Formulation

To select the best paths from the clusters, we combine informativeness $I(p_i^{C_j})$ and linguistic quality $LQ(p_i^{C_j})$ in an optimization framework. We maximize the following objective function:

$$F(p_1^{C_1}, \dots, p_K^{C_m}) = \sum_{j=1}^m \sum_{i=1}^K \frac{1}{T(p_i^{C_j})} \cdot I(p_i^{C_j}) \cdot LQ(p_i^{C_j}) \cdot p_i^{C_j} \quad (5)$$

Each $p_i^{C_j}$ represents a binary variable, that can take 0 or 1, depending on whether the path is selected in the final summary or not. In addition, $T(p_i^{C_j})$, the number of tokens in a

¹The model is available here: <http://www.keithv.com/software/giga/>. We used the VP 20K vocab version.

Table 1: Comparison of ROUGE scores on the DUC 2004 and 2005 datasets: Baselines, state-of-the-arts, our proposed methods and abstractive summarization system using MSC. “†” denotes the differences between [ILPSumm] and the baselines on the ROUGE scores are statistically significant for $p < 0.05$. We limit ROUGE evaluation to 665 bytes and 250 words for the 2004 and 2005 datasets, respectively.

DUC-2004			DUC-2005		
	ROUGE-2	ROUGE-SU4		ROUGE-L	ROUGE-SU4
Baselines			Baselines		
GreedyKL	0.08658	0.13253	Random	0.26395	0.09066
FreqSum	0.08218	0.12448	Centroid	0.32562	0.11007
Centroid	0.08139	0.12642	LexRank	0.33179	0.12021
TsSum	0.08068	0.12209			
LexRank	0.07796	0.12484			
State-of-the-arts			State-of-the-arts		
DPP	0.10079	0.14556	DUC Best	0.34764	0.10012
Submodular	0.09602	0.14227	LSA	0.26476	0.10806
RegSum	0.09712	0.13812	NMF	0.28716	0.11278
CLASSY04	0.09168	0.13250	KM	0.29107	0.10806
OCCAMSV	0.09420	0.13105	FGB	0.35018	0.12006
ICSIsumm	0.09585	0.13314	RDMS	0.35376	0.12297
CLASSY11	0.08912	0.12779			
Abstractive Systems					
$MD_{LexRank}^{Imp} + APO + MSC$ [Filippova, 2010]	0.09612	0.13911	$MD_{LexRank}^{Imp} + APO + MSC$	0.35589	0.12211
$MD_{LexRank}^{Imp} + MO + ILP$	0.09799	0.13884	$MD_{LexRank}^{Imp} + MO + ILP$	0.35281	0.12107
$MD_{LexRank}^{Imp} + APO + ILP$	0.10317	0.14218	$MD_{LexRank}^{Imp} + APO + ILP$	0.35342	0.12117
$MD_{CosSim}^{Imp} + MO + ILP$	0.09799	0.13884	$MD_{CosSim}^{Imp} + MO + ILP$	0.35661	0.12331
$MD_{CosSim}^{Imp} + APO + ILP$	0.10577	0.14215	$MD_{CosSim}^{Imp} + APO + ILP$	0.35577	0.12298
$MD_{DocsetSim}^{Imp} + MO + ILP$ [ILPSumm]	0.11992†	0.14765†	$MD_{DocsetSim}^{Imp} + MO + ILP$ [ILPSumm]	0.35772†	0.12411†
$MD_{DocsetSim}^{Imp} + APO + ILP$	0.11712	0.13567	$MD_{DocsetSim}^{Imp} + APO + ILP$	0.35679	0.12393

path, is also taken into consideration and the term $\frac{1}{T(p_i^{C_j})}$ assigns more weight to shorter paths so that the system can favor shorter informative sentences. We introduce several constraints to solve the problem. First, we ensure that a maximum of one path is selected from each cluster using Equation (6).

$$\forall j \in \{1, \dots, m\}, \sum_{i=1}^K p_i^{C_j} \leq 1 \quad (6)$$

We introduce Equation (7) so that we can prevent similar information (cosine similarity ≥ 0.5) from being selected from different clusters. In Figure 2, this constraint ensures that only one of the several possible paths mentioned in the example is included in the final summary as they contain redundant information.

$$\forall j, j' \in [1, m], i, i' \in [1, K] \subset C_j, C_{j'} \\ p_i^{C_j} + p_{i'}^{C_{j'}} \leq 1 \text{ if } \text{sim}(p_i^{C_j}, p_{i'}^{C_{j'}}) \geq 0.5. \quad (7)$$

4 Experimental Results

4.1 Dataset and Evaluation Metrics

We evaluated our approach on the DUC 2004 and 2005 datasets² on multi-document summarization. We use ROUGE (Recall-Oriented Understudy of Gisting Evaluation) [Lin, 2004] for automatic evaluation of summaries (compared against human-written model summaries) as it has been proven effective in measuring qualities of summaries and correlates well to human judgments.

²<http://duc.nist.gov/data.html>

4.2 ROUGE Evaluation

We proposed three document importance measures and two different sentence ordering techniques as described in Section 3. Hence, we have six different systems in total. To the best of our knowledge, no publicly available abstractive summarizers have been used on the DUC dataset. Therefore, we compare our system to MSC [Filippova, 2010] that generates a sentence from a collection of similar sentences using only syntactical information from the source sentences. In MSC, the input is a pre-defined cluster of similar sentences. Therefore, we compare our ILP based technique with MSC using the same set of input clusters obtained by our system. Table 1 shows the following ROUGE scores for our evaluation:

ROUGE-2: N -gram based ROUGE, where $N=2$, denotes the size of the sequence of words.

ROUGE-L (longest common subsequence),

ROUGE-SU4: skip-bigrams with unigrams.

The summaries generated by the baselines and the state-of-the-art extractive summarizers on the DUC 2004 data were collected from [Hong *et al.*, 2014]. ROUGE-2 and ROUGE-SU4 scores have been found to be highly correlated with human judgments [Nenkova and McKeown, 2011]. Therefore, we computed ROUGE-2 and ROUGE-SU4 scores of the other systems on the DUC2004 summaries directly using ROUGE. However, the system-generated summaries (baselines and state-of-the-arts) were not available for the DUC 2005 dataset. Hence, we used ROUGE scores of the various systems as reported in [Zheng *et al.*, 2014].

According to Table 1, all of the ROUGE scores obtained by our systems outperform all the baselines on both datasets. Hereafter, we refer to the best performing system as **ILP-Summ**. We perform paired t-test and observe that ILPSumm shows statistical significance compared to all the baselines. The summarization method using $MD_{DocsetSim}^{Imp}$ measure as

Table 2: Manual evaluation by 10 evaluators on *Informativeness* (Inf) and *Linguistic Quality* (LQ) of summaries. Average Log-likelihood scores (Avg.LL) from parser are also shown.

Type	Inf	LQ	Avg.LL
Human written	4.42	4.35	-129.02
Extractive (DPP)	3.90	3.81	-142.70
Abstractive (MSC)	3.78	2.83	-210.02
Abstractive (ILPSumm)	4.10	3.63	-180.76

the most informative document and ranked by majority ordering (MO) outperforms all of the other techniques. The document that has the highest similarity to the total content captures the central idea of the documents. The clustering scheme that works best with MSC is $MD_{LexRank}^{Imp}$ + APO. ILPSumm also outperforms the MSC-based method, i.e., our approach can generate more informative summaries by globally maximizing content selection from multiple clusters of sentences. In summary, content selection of our proposed abstractive systems work at par with the best extractive systems. **Discussion:** Our proposed system identifies the most important document, which is a general human strategy for summarization. The majority ordering strategy prioritizes clusters that contain sentences which should be mentioned earlier in a summary. Other systems tackle redundancy as a final step; however, we integrate linguistic quality and informativeness to select the best sentences in the summary using our ILP based approach.

We performed the rest of our experiments only on the DUC 2004 dataset as it has been widely used for multi-document summarization.

4.3 Manual Evaluation

We also determine readability of the generated summaries by obtaining ratings from human judges. Following Liu and Liu [2009], we ask 10 evaluators to rate 10 sets of four summaries on two different factors – *informativeness* and *linguistic quality*. The ratings range from 1 (lowest) to 5 (highest). All the evaluators have a good command of English and seven of them are native speakers. Evaluators were asked to rate the summaries based on informativeness (the amount of information conveyed) and linguistic quality (readability of the summary). We randomized the sets of summaries to avoid any bias.

The four summaries provided to the evaluators are human-written summary (one summary collected randomly from four model-summaries per cluster), extractive summary (DPP), abstractive summary generated using MSC (MSC) and abstractive summary generated using our ILP based method (ILPSumm). We asked each evaluator to complete 10 such tasks, each containing four summaries as explained above. We normalize ratings of different evaluators to the same scale. Table 2 shows the results obtained by manual evaluation. According to the judges, the linguistic quality of **ILPSumm** (3.63) is significantly better than that of **MSC** (2.83). Further, our summaries (ILPSumm) are more informative than DPP (3.90) and MSC (3.78). DPP is extractive in nature, hence linguistically, the sentences are generally more readable. To obtain a coarse estimate of grammatical-

Table 3: Example of DUC2004 abstractive summaries obtained using our approach (ILPSumm), extractive summaries (DPP) and and human-written summaries. Only a few initial sentences are displayed here.

Abstractive summary (ILPSumm): Hun Sen’s Cambodian People’s Party won 64 of the 122 parliamentary seats in July. Opposition ally Sam Rainsy charged that Hun Sen’s party has rejected allegations of intimidation and fraud. Hun Sen and Ranariddh are to form working groups this week to divide remaining government posts. But a deal reached between Hun Sen and his chief rival, Prince Norodom Ranariddh’s ally, Sam Rainsy.

Extractive summary (DPP): Ranariddh and Sam Rainsy have charged that Hun Sen’s victory in the elections was achieved through widespread fraud. Hun Sen said his current government would remain in power as long as the opposition refused to form a new one. Cambodian leader Hun Sen, who heads the CPP, has offered to share the legislature’s top job with the royalist FUNCINPEC party of Prince Norodom Ranariddh in order to break the impasse.

Human-written summary: Cambodian prime minister Hun Sen rejects demands of 2 opposition parties for talks in Beijing after failing to win a 2/3 majority in recent elections. Sihanouk refuses to host talks in Beijing. Opposition parties ask the Asian Development Bank to stop loans to Hun Sen’s government. CCP defends Hun Sen to the US Senate. FUNCINPEC refuses to share the presidency. Hun Sen and Ranariddh eventually form a coalition at summit convened by Sihanouk.

Abstractive Summary (ILPSumm): Lebanese Foreign Minister Kamal Kharrazi made the mediation offer Sunday, in a telephone conversation with his Syrian counterpart, Farouk al-Sharaa. Egyptian President Hosni Mubarak met here Sunday with Syrian President Hafez Assad to show Lebanon’s support for Syria and Turkey. In a show of force on Friday, Turkish troops were deployed this week on the Turkish-Syrian border to eradicate Krudish rebel bases.

Extractive Summary (DPP): Egyptian President Hosni Mubarak met here Sunday with Syrian President Hafez Assad to try to defuse growing tension between Syria and Turkey. The talks in Damascus came as Turkey has massed forces near the border with Syria after threatening to eradicate Kurdish rebel bases in the neighboring country. Egypt already has launched a mediation effort to try to prevent a military confrontation over Turkish allegations that Syria is harboring Turkish Kurdish rebels.

Human-written Summary: Tensions between Syria and Turkey increased as Turkey sent 10,000 troops to its border with Syria. The dispute comes amid accusations by Turkey that Syria helping Kurdish rebels based in Syria. Kurdish rebels have been conducting cross border raids into Turkey in an effort to gain Kurdish autonomy in the region.

ity, we also compute the confidence scores of the summaries using the Stanford Dependency Parser [Chen and Manning, 2014]. A language model assigns probabilities to sequence of words; in contrast, the confidence score of a parser assigns probabilities to grammatical relations. The values (the lower the magnitude, the better) are shown in the column Avg.LL. Avg.LL obtained by ILPSumm (-180.76) is better than that obtained by MSC (-210.02), indicating that the language model based linguistic quality estimation helps generate more readable summaries than the MSC method.

Table 3 shows a comparison of summaries from the different systems using the DUC 2004 dataset. As can be seen, the linguistic quality of the abstractive summaries (ILPSumm) is acceptable, and the content is well-formed and informative. Our ILP framework can combine information from various sentences and present a fairly well-formed readable summary.

Error Analysis: There is still room for improvement in the linguistic quality of the generated summaries. We analyzed the summaries that were given lower ratings than the other options on the basis of linguistic quality. Consider the following sentence in a system generated summary, which received low scores from human judges:

The U.N. imposed sanctions since 1992 for its refusal to hand over the two Libyans wanted in the 1988 bombing that killed 270 people killed.

As can be seen, the phrase “killed 270 people killed” is not coherent. The language model fails to identify such cases as the 3-gram sequences of *killed 270 people* and *270 people killed* are both grammatically coherent. In addition to a language model, we can also use a dependency parser to assign lower weights to paths that have redundant dependencies on the same nodes. Consider the following example:

The deal that will make Hun Sen prime minister and Ranariddh agreed to a government formed.

The last phrase – “a government formed”, is grammatically incoherent in the context of the sentence. Linguistically correct modifications could be – *a government being formed* or *a government formation*. In future work, we plan to address such issues of grammaticality using dependency parses of sentences rather than just adjacency relations when building the word-graph.

5 Conclusions and Future Work

We have proposed an approach to generate abstractive summaries from a document collection. We capture the redundant information using a simple yet effective clustering technique. We proposed a novel ILP based technique to select the best shortest paths in a word-graph to maximize information content and linguistic quality of a summary. Experimental results on the DUC 2004 and 2005 datasets show that our proposed approach outperforms all the baselines and the state-of-the-art extractive summarizers. Based on human judgments, our abstractive summaries are linguistically preferable than the baseline abstractive summarization technique. In future work, we plan to use paraphrasing techniques to further enhance quality of the generated summaries. We also plan to address phrase level redundancies to improve coherence.

References

- [Barzilay and Mckeown, 2005] Regina Barzilay and Kathleen R. Mckeown. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3):297–328, 2005.
- [Berg-Kirkpatrick et al., 2011] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly Learning to Extract and Compress. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT 2011)*, pages 481–490, 2011.
- [Chen and Manning, 2014] Danqi Chen and Christopher D Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014.
- [Erkan and Radev, 2004] Günes Erkan and Dragomir R Radev. LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22:457–479, 2004.
- [Filippova, 2010] Katja Filippova. Multi-Sentence Compression: Finding Shortest Paths in Word Graphs. In *Proc. of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 322–330, 2010.
- [Ganesan et al., 2010] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proc. of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, 2010.
- [Genest and Lapalme, 2010] Pierre-Etienne Genest and Guy Lapalme. Text Generation for Abstractive Summarization. In *Proc. of the 3rd Text Analysis Conference (TAC 2010)*, 2010.
- [Hong et al., 2014] Kai Hong, John M Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In *Proc. of the 9th International Conference on Language Resources and Evaluation (LREC’14)*, pages 1608–1616, 2014.
- [Knight and Marcu, 2002] Kevin Knight and Daniel Marcu. Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence*, 139(1):91–107, 2002.
- [Lin, 2004] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of the ACL-04 Workshop on Text Summarization Branches Out*, pages 74–81, 2004.
- [Liu and Liu, 2009] Fei Liu and Yang Liu. From Extractive to Abstractive Meeting Summaries: Can it be done by Sentence Compression? In *Proc. of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2009)*, pages 261–264, 2009.
- [Mani and Maybury, 1999] Inderjeet Mani and Mark T Maybury. *Advances in Automatic Text Summarization*. MIT Press, 1999.
- [Mehdad et al., 2013] Yashar Mehdad, Giuseppe Carenini, Frank W Tompa, and Raymond T NG. Abstractive Meeting Summarization with Entailment and Fusion. In *Proc. of the 14th European Workshop on Natural Language Generation*, pages 136–146, 2013.
- [Mihalcea and Tarau, 2004] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 404–411, 2004.
- [Nenkova and McKeown, 2011] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.
- [Oya et al., 2014] Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. A Template-based Abstractive Meeting Summarization: Leveraging Summary and Source Text Relationships. In *Proc. of the 8th International Natural Language Generation Conference (INLG 2014)*, pages 45–53, 2014.
- [Wan, 2008] Xiaojun Wan. An Exploration of Document Impact on Graph-Based Multi-Document Summarization. In *Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 755–762, 2008.
- [Wang and Cardie, 2013] Lu Wang and Claire Cardie. Domain-Independent Abstract Generation for Focused Meeting Summarization. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1395–1405, 2013.
- [Zajic et al., 2007] David Zajic, Bonnie J Dorr, Jimmy Lin, and Richard Schwartz. Multi-Candidate Reduction: Sentence Compression as a Tool for Document Summarization Tasks. *Information Processing & Management*, 43(6):1549–1570, 2007.
- [Zheng et al., 2014] Hai-Tao Zheng, Shu-Qin Gong, Hao Chen, Yong Jiang, and Shu-Tao Xia. Multi-document summarization based on sentence clustering. In *Neural Information Processing*, pages 429–436. Springer, 2014.