Readable and Coherent MultiDocument Summarization

by

Litton J Kurisinkel, Vigneshwaran Muralidharan, Vasudeva Varma, Dipti M Sharma

in

16th International Conference on Intelligent Text Processing and Computational Linguistics

Cairo, Egypt

Report No: IIIT/TR/2015/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2015

Readable and Coherent MultiDocument Summarization

Litton J Kurisinkel[1], Vigneshwaran M[1] ,Vasudeva Varma[2] , Dipti Misra Sharma[3]

International Institute of Information Technology / Hyderabad 500032, India
litton.jKurisinkel@research.iiit.ac.in, vigneshwaran.m@research.iiit.ac.in
vv@iiit.ac.in, dipti@iiit.ac.in

Abstract. Extractive summarization is the process of precisely choosing a set of sentences from a corpus which can actually be a representative of the original corpus in a limited space. In addition to exhibiting a good content coverage, the final summary should be readable as well as structurally and topically coherent. In this paper we present a holistic, multi-document summarization approach which takes care of the content coverage, sentence ordering, maintenance of topical coherence, topical order and inter-sentence structural relationships. To achieve this we have introduced a novel concept of a Local Coherent Unit(LCU). Our results are comparable with the peer systems for content coverage and sentence ordering measured in terms of ROUGE and τ score respectively. The human evaluation preference for readability and coherence of summary are significantly better for our approach vis a vis other approaches. The approach is scalable to bigger real-time corpus as well.

1 Introduction

Automated text summarization enables the reader of the summary to understand the essence of information contained in a big corpus of documents without going through the entire set. Extractive summarization techniques try to achieve this by selecting a proper subset of sentences from the corpus, which constitute the summary. Most of the techniques adopted for extractive summarization can be understood to perform three basic steps.

1. Create an intermediate representation for the target text such that the key textual features within are captured.
2. Using the generated intermediate representation, assign scores for individual sentences within the text.
3. Finally select a set of sentences which maximizes the total score as the summary of the target text.

Possible intermediate representations are created by Topic Signatures, Word frequency count approaches, Latent Space Approaches using matrix factorization or Bayesian Approaches. In almost all of the approaches the smallest linguistic unit which is to be scored and selected for summarization is a sentence. Most of the prevalent scoring functions consider quantifying the priority of the sentence for better content coverage. In these approaches, the output set of sentences are later fed to a distinct sentence-ordering component which reorders the sentences. By the time a precise subset of sentences are chosen, most of the information related to the inter-sentential structural dependency are lost. Most of the re-ordering algorithms can achieve only a topical order leaving behind the possibility of out-of-context sentence usage as given below.

e.g. Nevertheless this object pulls everything which enters its event horizon.

The above sentence might secure a high score in terms of topical significance if the corpus is on *Black hole* but can still result in an out-of-context sentence placement. This can cause an incoherent reading or sometimes result in an erroneous inference. We propose a novel concept called as a Local Coherent Unit(LCU) which enforces a contextual constraint for sentence extraction. *An LCU is a unit of text containing sequence of sentences such that, excluding the first sentence, every subsequent sentence within the unit has an explicit discourse dependency with the preceding sentence.* The explicit discourse dependency can be of any type such as an event adverbial related to previous sentence, anaphoric reference, deictic pointers to previous entities etc. These are realized in sentences as structural dependency cues.

We discuss about the relevant works done on summarization and sentence ordering in Section 2. In section 3 we discuss an overview of all the components of our system and their organization. The section comprises of the subsection 3.1 which explains about a stand-alone component which identifies LCUs. Then we elaborate about the topic modelling, document merging, topic segmentation in subsections 3.2, 3.3 and 3.4 respectively. The role of topic segmentation and LCU in our summarization process is explained in subsections 3.5 and 3.6. Finally the experimental results are discussed in section 4.

2 Related Work

Extensive work has been done on extractive summarization which tries to achieve a proper content coverage by scoring and selection of sentences. All these previous works seek the help of a second component to re-order the set of extracted sentences. Most of the extractive summarization researches aim to increase the total salience of the sentences while reducing redundancy. Approaches include the use of Maximum Marginal Relevance [1], Centroid-based Summarization [2], Summarization through Keyphrase Extraction [3] and Formulation as Minimum Dominating Set problem [4]. Graph centrality has also been used to estimate the salience of a sentence [5]. Approaches to content analysis include generative topic models [6], [7],[8] and Discriminative models [9].

ILP2 [10] is a system that uses Integer Linear Programming(ILP) to jointly optimize the importance of the summary's sentences and their diversity (non-redundancy), while also respecting the maximum allowed summary length. They use a Support Vector Regression model to generate a scoring function for the sentences. Woodsend and Lapata [11] arrived at a scoring function which holds linear components to quantify the salience of bi-grams, salience of parse tree nodes and a component based on a language model which penalises the unlikely sentences. An approach based on the distribution of some important concepts in the summary was done by [12]. The concepts are bi-grams in the corpus to be summarised. They formulated an ILP objective function in the space of candidate summaries that maximizes the total concept weight score of the summary to be chosen.

Takamura and Okumura [13] have treated multidocument summarization as a maximum concept coverage problem with knapsack constraint(MCKP). They have also exploited the possibility of decoding algorithms in solving MCKP in the summarization task. Lin and Bilmes[14] formulated summarization as a sub-modular function maximization problem in the possible set of candidate summaries with due respect to the space constraint. All the above methods have concentrated on content coverage but have the drawback of out-of-context sentence usages.

As far as sentence ordering is concerned, Li et al. [15] used context inference to achieve better sentence ordering while McKeown et al[16] used majority ordering algorithm to sort sentences. Lapata [17] provided an unsupervised probabilistic model for sentence ordering while Ji et al [18] used a cluster adjacency based approach. Though the sentence ordering approaches can achieve a topical order of sentences, the local structural relations of the sentences are never captured.

The work which pioneered a holistic approach towards multi-document summarization by bringing sentence selection and coherence under a single umbrella is G-Flow by [19]. They built a graph which stored discourse relations with proper edge weights to quantify coherence. This value was linearly combined along with salience and redundancy in the scoring function of sentences to formulate multi-document summarization as a constraint optimization problem.

The system has taken into consideration the readability of the extracted sentences in output summary by quantifying its coherence by means of discourse graph. With the increase in corpus size, the space complexity of generating discourse graph with large ‘n’ is of the order O(n^2). The optimization function in this case cannot take a greedy approach for inducing coherence while selecting and discarding sentences for output summary. This is because the coherence is measured for the whole chosen candidate summary and there is no way to greedily choose potentially coherent sentences individually. As per [14], having the objective function as a submodular non-decreasing function can incorporate a greedy approach that guarantees a solution at the most as good as the best solution with a factor of 0.632. Hence we have used an LCU based submodular non-decreasing function in our summary extraction step while letting LCU ensure the required readability and coherence.

3 Our Approach

More than sentence scoring and content selection which aim only at content coverage, a summary should be readable and intelligible to a human reader without any previous knowledge on the content of the corpus. A summary which is topically uniform may not capture different topical aspects of the corpus and a summary which is too diverse can take the form of a short note which can only be understood by a person having prior knowledge about the content of the corpus. So an optimal topical coherence, which conveys a gist of the various topics of the corpus, packed within the constraints of target summary size along with a proper sentence order needs to be achieved.

The main intuition behind the choice of our approach begins with a crucial question about the linguistic nature of a text. *Is text a bag of words every time?* It need not be so because, for instance, the first sentence taken from each paragraph of a document can account for a reliable summary of the whole document. Psycholinguistic studies suggest that local coherence plays a vital role in inference making during the reading [20]. Local coherence is undoubtedly necessary for global coherence and has received considerable attention in Computational Linguistics. (Marcu [21], Kintsch et al[22], Althaus et al[23], Karamanis et al [24]).

To handle the explicit structural coherence created by the sentences in a document, we conceptualized a notion called as Local Coherent Unit(LCU). An LCU is a unit of text containing a sequence of structurally dependent adjacent sentences in a document. The LCU will be used as a basic unit of processing for summary extraction which implicitly imposes restriction of out-of-context sentence usage and hence more readable. The next section describes in more detail what is meant by structural dependency between sentences, how to identify Local Coherent Units etc.

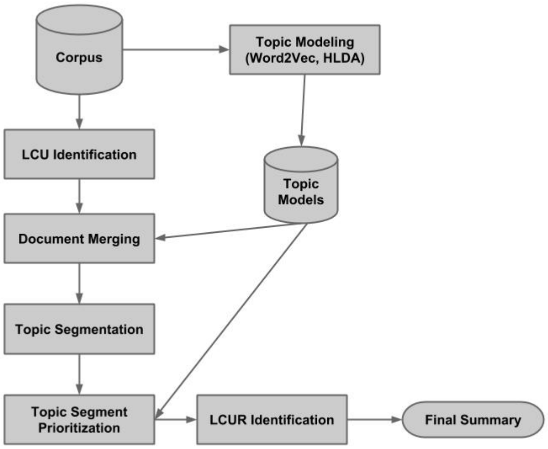

Fig. 1. System Architecture

The architecture of the system is shown above. A brief explanation of the above system architecture is as follows:

1. Given a corpus of multiple documents, a complete, non-overlapping set of LCUs in every one of those documents are identified. Now the basic unit of processing for every document is a sequence of LCUs, not sentences.
2. A HLDA tree which represents the latent topic structure based on term distribution is created for the entire corpus by considering paragraphs as documents i.e the entire set of paragraphs is the input corpus for HLDA
3. Word2Vec tool takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The word vector so generated contains top 'n' list of words closer to the current word based on cosine distance between them. Such a Word2Vec model is created for the entire corpus.
4. Word2Vec vectors of all the terms in an LCU are calculated. Now the mean of all the word vectors in the LCU is considered to represent the LCU itself in meaning space. Any two LCUs can be compared for topical similarity using cosine similarity of their mean word vectors. Two LCUs can likewise be compared for parent-child hierarchy of their terms using HLDA tree.
5. Using the above topic models, a total topical ordering of the LCUs in the entire corpus is performed. We call this step as document merging. The merged document now is the single output containing sequence of all LCUs in the corpus in topical order.
6. A topic segmentation is performed explicitly on the merged document to identify topic boundaries thus creating segments of topics. Again a topic segment is just a sequence of LCUs in the merged order since we have not disturbed it in topic segmentation. This step is required on top of the merged document in order to scale up the coherent summary extraction approach for bigger corpus with multiple documents containing larger text.
7. Final summary is something that has to be extracted from diverse non-redundant topics while the sentences extracted have to be readable in the sequence of extraction.
8. For this, we find topic priority of each topic segment to identify its contribution to the final summary. Finally noise-free representatives of LCU are chosen from every topic segment proportional to their priority such that the extracted summary is optimal.

3.1 Local Coherent Units Identification

Every document in a corpus is a set of sentences which together form a discourse. For summarization it is necessary to retain the discourse level relations between sentences and make use of those while extracting content for summary. Typically discourse level relations can be identified by a discourse parser developed based on Rhetorical Structure Theory[25], Penn Discourse Tree Bank[26]. Usefulness of discourse indicators for content selection in summarization has been explored [27] and the robustness of structure information in the identification of importance of a text has been discussed.

However we observed that instead of explicitly modelling the discourse relations between sentences, representation of document as a set of Local Coherent Units helps

Algorithm 1 LCU Identification Algorithm

```
Require: $DEP\_PARSE\_TREES, DEP\_CUES, ARG\_CUES$
  for each i ∈ $DEP\_PARSE\_TREES$ in the document do
    $cue\_dep\_stack$ ← ”ROOT”
    $cue\_arg\_stack$ ← ””
    flag ← 0
    while $cue\_dep\_stack \neq$ ”” do
      temp ← $cue\_dep\_stack$.pop()
      $cue\_arg\_stack$ ← searchRightArg(i,temp)
      while $cue\_arg\_stack \neq$ ”” do
        temp2 ← $cue\_arg\_stack$.pop()
        if temp2 is an entry in $DEP\_CUES$ then
          flag ← 1
          Break from immediate while loop which checks $cue\_arg\_stack$
        else
          $cue\_arg\_stack$.push(temp2)
        end if
        if flag==1 then
          ADD THE CURRENT SENTENCE i TO EXISTING LCU
        else
          ADD THE CURRENT SENTENCE i TO NEW LCU
        end if
      end while
    end while
  end for
```

capture the inter-sentence structural dependencies that can be best utilized for summary extraction incrementally. The structural dependency is defined in terms of a set of linguistic cues obtained from the dependency tree for every individual sentences. The identification of these local coherent units can also be done for languages which do not have a fully developed discourse parser and hence we decided to apply this strategy as a component in multi-document summarization.

Initially we start with one empty LCU. Once we have the parsed output of the current document from a dependency parser[1], the decision that has to be taken for each sentence is whether it belongs to a previous LCU or begins a new one. Some linguistic cues(’nsubj’ modified by demonstratives etc.) were used to decide whether the current sentence has a structural dependency on previous sentence. If such a dependence exists the current sentence is added to the existing LCU. If not, a new local coherent unit is created and the sentence is added to it. This is continued till the end of the document and as a result the document will be segmented as a series of LCUs. By processing all the documents of the corpus in the same manner, we get a representation now where the documents are understood as a series of LCUs which can be used later for applying

[1] Stanford Dependency parser Version 3.3.1
http://nlp.stanford.edu/software/lex-parser.shtml#Download

statistical methods.

Example for a local coherent unit is given below.

> e.g. *Black holes are intriguing ideas.* However *they are not likely to account for much dark matter.*

Algorithm 2 Search Right Arguments Recursively

```
Require: $DEP\_TREE, CURR\_DEP\_RELATION$
  $arg\_stack \leftarrow \{\}$
  for All lines throughout the parse tree do
    if Right argument of $CURR\_DEP\_RELATION$ occurs as Left argument in one of $DEP\_CUES$ then
      Add the Right argument to $arg\_stack$
    end if
  end for
  return $arg\_stack$
```

The above LCU has two sentences in it. *However* is a CC in the main clause of the second sentence which shows structural dependency with first sentence. By a simple set of rules which take the cues tabulated in Table 3.1 and using a finite set of arguments for such dependency relations mentioned below LCU can be formed.

Table 1. Cue dependencies called as DEP_CUES for LCU Identification

Relation	Meaning	Examples
nsubj	Subject of main clause	He,She
dobj	Direct object of main verb	He,They
det	Determiner,Demonstratives	This,The
mark	Subordinate marker	that, if
nsubjpass	Subject of passive	He,It
advmod	Adverbial modifier	Still, thus
CC	Coordinate conjunction	And, Yet

The dependency parsed output of an entire document is taken as the input and the Algorithm 1 is run to get the sequence of LCUs identified within the document. For executing the algorithm we need set of cue dependencies and a set of cue argument values which trigger structural dependency with preceding sentence are finite in number. We have chosen the below list of dependencies and arguments.

1. Dependency cues DEP_CUES *root, nsubj, dobj, det, mark, nsubjpass, advmod, cc*

2. Argument cues ARG_CUES - *All third person pronouns and their inflected forms, Demonstratives, 20 adverbs which act as explicit discourse connectives such as so, thus, still etc*

There are many discourse markers and information structure cues English. But as a preliminary approach we have chosen the above ones as these exhibited a reasonable coverage of correct LCU identification as shown in 4. The approach can be extended by using other discourse relations as well. The dependency parse tree of a line consists of entries of the form Dependency_Reln(leftarg, rightarg). The Algorithm 2 tries to find recursively all right arguments which are present as left argument for some cue dependency relations in the parse tree. All such identified right arguments are added to the stack cue_arg_stack and returned.

3.2 Word2Vec and HLDA Modelling

Reliable topic models created for the corpus can enhance the process of automated summarization. Topic Hierarchy of the corpus is identified by creating an HLDA model[2] [28]. The paragraphs in a document hold the explicit topic-wise organization of text conceived by the author. So the paragraphs hold a sufficient amount of prior information about the topic-term distribution. Therefore for the purpose of HLDA tree creation, we treat paragraphs as documents and the entire set of paragraphs as input corpus for HLDA. As paragraphs are usually fine grained on a few topics in a well written document, variable α for HLDA which corresponds to the prior for per document(paragraph) topic distribution is kept at a very low value. We have created a Word2Vec model[3] to find the semantic similarity between any two text units. Each word is vectorised by choosing top 'n' similar words from Word2Vec and their corresponding similarity values. *To vectorise a text unit we take the mean vector of all word vectors in the text unit.*

3.3 Document Merging

Once each document is represented as a sequence of LCUs, each LCU in the corpus is assigned a corpus level Id . Local coherent units are relatively much larger than a sentence and hold enough information to decide their topical identity. The task of summarization becomes easier once we could merge these documents into a topically coherent document without violating inter-sentence structural relationships. As LCUs already hold inter-sentence structural relationships, arriving at a sequence of corpus-level LCU ids which exhibits maximum topical order and coherence can result in the best merged document that is possible. *For this purpose we utilise the HLDA and Word2Vec model created for the corpus.* Document merging can be framed as an optimization problem where we maximize the function given by the Equation (1) in the space of all possible

[2] https://github.com/chyikwei/topicModels
[3] http://radimrehurek.com/gensim/models/Word2Vec.html

sequences of LCUs.

$$\begin{aligned}\mathbf{Q}(\mathbf{Z}) = \sum_{i=1}^{N-1} \mathbf{100} * \mathbf{W2V}(\mathrm{LCU}_i, \mathrm{LCU}_{i+1}) + \\ \mathbf{DD}(\mathrm{LCU}_i, \mathrm{LCU}_{i+1}) \; - \; \mathbf{DD}(\mathrm{LCU}_{i+1}, \mathrm{LCU}_i)\end{aligned} \tag{1}$$

where

Z→ *Possible sequence of LCUs in the corpus*
N → *Total number of LCUs in the corpus.*
W2V → *Word2Vec cosine similarity*
LCU_i → *i^{th} LCU in the sequence.*
DD → *Function call to Algorithm4*

Algorithm(4) quantifies the extent upto which the topics dealt in the LCU2 belong to the sub-topic category of LCU1. First term in the above Equation (1) brings all coherent units which deal with semantically similar topics together. Second and third terms arrange them in a proper topic to subtopic order. Since the framing of document merging as an optimization problem can be costly for real-time usage, we have used a greedy algorithm which approximates the function in the Equation 1. For the convenience of greedy approximation at each step we have reframed the 2 as below.

$$\begin{aligned}\mathbf{F}(\mathbf{LCU1}, \mathbf{LCU2}) = \mathbf{100} * \mathbf{W2V}(\mathrm{LCU}_1, \mathrm{LCU}_2) + \\ \mathbf{DD}(\mathrm{LCU}_1, \mathrm{LCU}_2) \; - \; \mathbf{DD}(\mathrm{LCU}_2, \mathrm{LCU}_1)\end{aligned} \tag{2}$$

The Algorithm(3) uses the above function F in Equation(2). It takes two documents, arranges the LCUs from two documents in the optimum order and returns the merged document. In the algorithm 4 *H* stands for the height of HLDA tree, *x.level* is the level of topic node in the HLDA tree to which the term x belongs with a maximum chance, AncestorNodes(i) is the set of all ancestor nodes of the node in the HLDA tree to which term i belongs with maximum chance, DescendentNodes(i) is the set of all descendent nodes of the node in the HLDA tree to which the term i belongs with maximum chance. Overall we are trying to find the insertion position of coherent unit in the document which maximizes the above function given by Equation 2. The merge algorithm starts with first two documents to form a single merged document. This merged output is further merged with the third document and the process incrementally continues until all the documents in the corpus are merged into a single structure.

3.4 Linear Topic Segmentation Using Affinity Propagation Algorithm

The larger merged document formed as a sequence of all local coherent units in the corpus is linearly segmented into topic segments which contain more than one local coherent units. Each topic segment exhibits a high level of topic uniformity. We employ the implementation of ‘Linear text segmentation by affinity propagation’ by [29] for segmenting the merged document.

Algorithm 3 Document Merging Algorithm

```
Require: Doc1, Doc2, HLDA model
  maxScoringPair $\leftarrow$ (0,0)
  currentFnValue $\leftarrow$ 0
  maxvalue $\leftarrow$ 0
  for each LCU i $\epsilon$ Doc2 do
    for each LCU j $\epsilon$ Doc1 do
      currentFnValue $\leftarrow$ F(i,j)
      if currentFnValue $>$ maxvalue then
        maxScoringPair $\leftarrow$ (i,j)
        maxvalue $\leftarrow$ currentFnValue
      end if
      currentFnValue $\leftarrow$ F(j,i)
      if currentFnValue $>$ maxvalue then
        maxScoringPair $\leftarrow$ (j,i)
        maxvalue $\leftarrow$ currentFnValue
      end if
    end for
    if maxScoringPair = (i,j) then
      insert i above j in Doc1
    end if
    if maxScoringPair = (j,i) then
      insert i below j in Doc1
    end if
  end for
  return Doc1
```

Affinity propagation algorithm for segmentation receives a set of pairwise similarities between data points and decides the topic segment boundaries and segment centres. A segment centre is a data point which best describes all other data points within the segment. Data points in our merged document are local coherent units. The similarity measure to be supplied to the topic segmentation algorithm is calculated by the cosine similarity between mean word vector of local coherent units. Another important parameter for topic segmentation algorithm is the set of preference values which represents the *a priori* belief of each data point to become a segment centre. Preference value of a local coherent unit during linear topic segmentation is calculated as the mean cosine similarity between k neighbouring local coherent units in the merged document[4].

3.5 Prioritization of each Topic Segment for Summarization Process

As an analogy to understand the topic segments, it can be seen that reduction of an image from a richer dimension to lower dimensions can cause certain objects in the image to get eliminated and some among them to get abstracted. Topic segments in the merged document are analogous to the objects in the high resolution image. We prioritize the topic segments and identify their level of participation in the final summary. During this

[4] The value of 'k' is experimentally optimized.

Algorithm 4 Descendent Level Difference Calculation

```
Require: LCU1,LCU2,HLDA model
  levelDiff $\leftarrow$ 0
  for each term i $\epsilon$ LCU1 do
    for each term j $\epsilon$ LCU2 do
      if j $\epsilon$ Descendents(i) then
        levelDiff $\leftarrow$ levelDiff+H-(j.level-i.level)
      end if
    end for
  end for
  return levelDiff
```

process some among them get abstracted and some get eliminated to generate a coherent summary that is best conveyable within the allowed summary space. The priority of the topic segment *T* is decided by the Equation *(3)* below.

$$\mathbf{P(T)} = \omega 1 * \mathbf{SDI(T)} + \omega 2 * \mathbf{G(T)} \tag{3}$$

where P(T) refers to the priority of the topic segment T, SDI(T) refers to Shannon's diversity index of the topic segment T and G(T) refers to the generality of the topic segment.

The first component decides the information content of the topic segment where the second one decides the generality of information contained. The two terms SDI Shannon's diversity index and Generality are given by

$$\mathrm{SDI(T)} \ = \sum \mathrm{p}_i \ \ln(\mathrm{p}_i) \ \forall \ \mathrm{term\ i} \in \mathrm{T} \tag{4}$$

where $\mathrm{p_i}$ is the normalized frequency of term *i* in *T*.

$$\mathrm{G(T)} = \sum \frac{(\mathrm{H} - (\mathrm{t.level})}{\mathrm{n} * \mathrm{H}} \atop \forall \ \mathrm{terms\ t} \ \in \ \mathrm{topic\ segment\ T} \tag{5}$$

where *H* is the height of HLDA tree, *n* is the total number of terms in topic segment *T*. The segment priorities calculated from Equation (3) are normalized between 0 and 1. The proportional contribution *binsize$_i$* of each topic segment *i* for the final summary is calculated as

$$\mathrm{binsize_i} \ = \ \mathrm{P(i)} * \mathrm{Targeted\ summary\ size} \tag{6}$$

3.6 Summarization of each Topic Segment

Topic segments are summarized by selecting noise-free representatives of LCUs till the allotted bin size of the topic segment is exhausted. These noise-free representatives are called LCURs.

We have identified Local Coherent Units to avoid an out-of-context sentence usage. But when a non-pruned local coherent unit which is relatively larger than a sentence is directly included as a representative in the final write-up, it can result in a noisy summary in terms of relevance and generality. We have to extract a noise-free combined representative of a local coherent unit without disturbing the structural dependency that is preserved within an LCU.

For this purpose we consider that every sentence in an LCU is depending on all of its previous sentences. So the possible candidate representatives of an LCU containing sentences S1, S2, S3 and S4 are $\{S1\}$, $\{S1,S2\}$, $\{S1,S2,S3\}$, $\{S1,S2,S3,S4\}$ and we call them local coherent unit representatives(LCUR). This combination was currently chosen to ensure that even after an LCU is pruned the structural dependency between the resulting sentences should be retained. If we choose a combination such as S1,S3 from within an LCU, the possible structural dependency of sentence S3 with S2 would be lost. Such pruning would defeat the purpose of having an LCU.

We use a variant of greedy version of Maximum Word Coverage Algorithm[13] for summarization. We greedily choose the LCUR with highest normalized score of Equation (7) as the candidate representative of the given LCU. At any given instance if a candidate LCUR is chosen, all other LCURs from the same local coherent unit are discarded.

$$\begin{gathered}\mathrm{SF(LCUR)} = \lambda 1 * \sum \frac{(\mathrm{TF\text{-}IDF(w_i)})}{\mathrm{n}} + \\ \lambda 2 * \mu + \lambda 3 * (\mathrm{TZ}) + \lambda 4 * (\mathrm{FZ});\end{gathered} \tag{7}$$

where

$\mathrm{W_{set}} \rightarrow$ *set of words chosen so far in summary at the current iteration of greedy algorithm.*
Word $w_i \in$ *LCUR and* $w_i \notin W_{set}$.
$n \rightarrow$ *Number of words in LCUR.*
$\mu \rightarrow$ *Average size of sentences in LCUR*

$TZ \rightarrow \frac{\text{Size of LCU to which LCUR belongs}}{\text{Topic segment size}}$.
$FZ \rightarrow \frac{\text{Bin size allotted for the topic segment}}{\text{SizeofLCUR}}$

Topic segments are treated as documents to calculate TF-IDF of words w_i. The second term in Equation (7) gives a small priority for the LCURs containing longer sentences while the third term in the function gives a slight priority for an LCUR of a longer LCU. FZ term encourages the selection of LCURs from different LCUs. In each iteration the Greedy algorithm selects the maximum scoring LCUR and continues to include them till the topic segment summary *equals or just crosses the segment's allotted bin size*. We do not form topic segment summaries less than the allotted bin size in order to avoid an aggregate deficit in the targeted summary.

We repeat this for all topic segments and aggregate the set of LCURs from each of the topic segment summaries. The total size of the summary formed out of these bunch of aggregated LCURs could slightly exceed the targeted summary length(Because for every topic segment we chose *topic summary of length* $>=$ *binsize*). Now the same greedy algorithm with the above mentioned scoring function is applied on the aggregated LCURs such that the final summary does not exceed the targeted summary size in number of bytes.

In the objective function used above, the average TF-IDF score for an LCUR is calculated only for the words which are uncovered by the summary till the current iteration of greedy algorithm. This avoids the explicit usage of diversity measure. As this component of the function is submodular and non-decreasing and all other components have constant values for an LCUR at any stage of iteration, the function SF is submodular and non-decreasing. The LCURs extracted are arranged in the same order in which they occur in the merged document.

4 Experiments and Results

Different components of the system such as Local Coherent Unit Identification, Document Merging, Sentence Ordering and Content Coverage are evaluated using DUC 2004 Task2 Dataset [5] as it contains documents of sufficient size for HLDA modelling[28]. As proper sentence ordering is a consequent of document merging, both need not be tested separately. DUC 2004 contains 50 cluster of documents each containing 10 documents and 4 manual summaries.

4.1 Content Coverage

We have taken DUC 2003 as our development set on which function weights of Equation (3) and Equation (7) and HLDA parameters are optimized using grid search. HLDA parameters α, β and γ are optimized for achieving better ordering of merged document for each cluster in terms of Kendall's τ (Lebanon, 2002) measure.The weights of Equations (3) and (7) are optimized for achieving maximum ROUGE score [30] with reference summaries. The major systems which has reported results on DUC 2004 dataset for *Content coverage* are [31] [13], [14] and G-FLOW [19]. We have chosen domain independent generic features for summarization and got comparable results in terms of ROUGE-1 recall and F-measure values. We have tested the system with and without topic segmentation. Results of content coverage are tabulated in Table 2.

When the system was tested without employing the topic segmentation thereby treating the whole merged document as one topic segment, the content coverage was high but readability and coherence was relatively lesser. With topic segmentation, it can be seen that the content coverage was comparable while at the same time sentence ordering is improved.

[5] http://www-nlpir.nist.gov/projects/duc/data/2004_data.html

Table 2. Content Coverage Results

Approach	Rouge-R	Rouge-F
Nobata&Sekine(2004)	30.44	34.36
G-FLow(2013)	37.33	37.43
Our system (Without Topic Segmentation)	37.65	37.70
Our system (With Topic Segmentation)	36.42	36.65
Takamura&Okumura (2009)	38.50	-
Lin & Bilmes(2011)	39.35	38.90

4.2 Sentence Ordering

In addition to content coverage, we have compared the results of our approach with the results of existing sentence ordering approaches of [15], [16], [17] and [18].

As the reference summaries of DUC 2004 Task2 contained human framed sentences for each sentence we have chosen the offset of LCU in the merged document which has maximum cosine semantic similarity with the sentence to represent its position with respect to our system. Offsets of sentences in a reference summary has to be in increasing order. The difference of the actual order with a desired increasing order is measured using Kendall's τ. Our average measure for the corpus is comparable with other peer systems for sentence ordering. The results are tabulated in table 3.

Table 3. Sentence Ordering Results

Approach	Kendall's τ
McKeown et al.	0.143
Lapata et al.	0.144
Our System	0.387
Ji et al.	0.415
Li et al.	0.432

4.3 LCU Identification Accuracy

To find out how accurate the identified Local Coherent Units are, we compare it against the manually identified local coherent units on the test corpus. We had chosen a collection of 5 sample documents from DUC corpus for which we had manually identified the Local coherent units. The percentage accuracy of proper identification of local coherent unit is measured as the number of edit operations required to align the system output with the ideal manual annotation of LCUs. The number of sentences moved/split during this alignment is used to calculate the accuracy.

$$\mathrm{Accuracy} = (1 - (\mathrm{EC}/\mathrm{N})) * 100 \tag{8}$$

Here *EC* is the number of edit operations required to match the system output LCU with human-identified LCUs and *N* is the number of sentences in the document. Our overall accuracy for identifying LCUs was 78.26%. Details are tabulated below in Table 4.

Table 4. Local Coherent Unit Identification Accuracy

DocNo.	Edits	Sentences	Accuracy%
1	10	52	80.76
2	35	134	73.88
3	16	68	76.47
4	14	55	70.91
5	6	56	89.29

4.4 Overall Summary Quality

In order to test the overall readable quality and coherence of our summary we performed a readability evaluation experiment in which 6 participants were given pairs of summary - one generated by state-of-the art summarization system by G-Flow and the other generated by our system - for all the clusters in the DUC 2004 dataset. The 6 evaluators were the research students of Computational Linguistics, who could effectively decide the summary quality in terms of readability and coherence. The two candidates of summary pair were shown in random order and the evaluators had to choose which candidate summary rated better. If the evaluator was ambiguous about his choice he could stay indifferent and mark the rating as ‘ambiguous’. As seen below, the preference for our system is more than the G-Flow.

Table 5. Overall Summary Preference

Our approach	G-Flow	Ambiguous
47%	41%	12%

We also have compared the summary quality of the system with and without performing topic segmentation. The overall quality of the summary was higher when topic segmentation is performed.

Table 6. Summary preference within our approach

With topic segmentation	Without topic segmentation	Ambiguous
60%	33%	7%

5 Conclusion

Treating summarization as a content coverage optimization problem by selecting individual sentences as candidates can achieve a flexible content coverage but may result in incoherent summary. We have treated summarization not just as an optimization of content coverage but also have retained the inter-sentence structural relationships at the level of LCU intact. For now, we assumed a linear structure for the local coherent unit(LCU) as a starting point for the approach.

Going forward we can incorporate a graphical structure for a local coherent unit which gives a more noise-free LCUR. We have used a variant of concept coverage algorithm without any corpus dependent features which makes this approach general enough for a domain-independent summarization. The merits of HLDA topic model can be better realized for real-time bigger documents which have better paragraph organization structure thus improving ordering of sentences.

References

1. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1998) 335–336
2. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40**(6) (2004) 919–938
3. Qazvinian, V., Radev, D.R., Özgür, A.: Citation summarization through keyphrase extraction. In: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics (2010) 895–903
4. Shen, C., Li, T.: Multi-document summarization via the minimum dominating set. In: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics (2010) 984–992
5. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research (2004) 457–479
6. Haghighi, A., Vanderwende, L.: Exploring content models for multi-document summarization. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 362–370
7. Celikyilmaz, A., Hakkani-Tur, D.: A hybrid hierarchical model for multi-document summarization. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 815–824
8. Li, P., Wang, Y., Gao, W., Jiang, J.: Generating aspect-oriented multi-document summarization with event-aspect model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011) 1137–1146
9. Aker, A., Cohn, T., Gaizauskas, R.: Multi-document summarization using a* search and discriminative training. In: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics (2010) 482–491
10. Galanis, D., Lampouras, G., Androutsopoulos, I.: Extractive multi-document summarization with integer linear programming and support vector regression. In: COLING, Citeseer (2012) 911–926

11. Woodsend, K., Lapata, M.: Multiple aspect summarization using integer linear programming. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics (2012) 233–243
12. Berg-Kirkpatrick, T., Gillick, D., Klein, D.: Jointly learning to extract and compress. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 481–490
13. Takamura, H., Okumura, M.: Text summarization model based on maximum coverage problem and its variant. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 781–789
14. Lin, H., Bilmes, J.: 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, Association for Computational Linguistics (2011) 510–520
15. Li, P., Deng, G., Zhu, Q.: Using context inference to improve sentence ordering for multi-document summarization. In: IJCNLP. (2011) 1055–1061
16. McKeown, K., Hatzivassiloglou, V., Barzilay, R., Schiffman, B., Evans, D., Teufel, S.: Columbia multi-document summarization: Approach and evaluation. (2001)
17. Lapata, M.: Probabilistic text structuring: Experiments with sentence ordering. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Association for Computational Linguistics (2003) 545–552
18. Donghong, J., Yu, N.: Sentence ordering based on cluster adjacency in multi-document summarization. IJCNLP 2008 (2008) 745–750
19. Christensen, J., Mausam, S.S., Soderland, S., Etzioni, O.: Towards coherent multi-document summarization. In: HLT-NAACL, Citeseer (2013) 1163–1173
20. McKoon, G., Ratcliff, R.: Inference during reading. Psychological review **99**(3) (1992) 440
21. Marcu, D.: The theory and practice of discourse parsing and summarization
22. Foltz, P.W., Kintsch, W., Landauer, T.K.: The measurement of textual coherence with latent semantic analysis. Discourse processes **25**(2-3) (1998) 285–307
23. Althaus, E., Karamanis, N., Koller, A.: Computing locally coherent discourses. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics (2004) 399
24. Karamanis, N., Poesio, M., Mellish, C., Oberlander, J.: Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics (2004) 391
25. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse **8**(3) (1988) 243–281
26. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A.K., Webber, B.L.: The penn discourse treebank 2.0. In: LREC, Citeseer (2008)
27. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics (2010) 147–156
28. Griffiths, D., Tenenbaum, M.: Hierarchical topic models and the nested chinese restaurant process. Advances in neural information processing systems **16** (2004) 17
29. Kazantseva, A., Szpakowicz, S.: Linear text segmentation using affinity propagation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2011) 284–293

30. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. Volume 8. (2004)
31. Nobata, C., Sekine, S.: Crl/nyu summarization system at duc-2004. In: Proceedings of DUC. (2004)