GENERATING SUMMARIES USING SENTENCE COMPRESSION AND STATISTICAL MEASURES

Shouvik Roy

Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

ABSTRACT

In this paper, we propose a compression based multi-document summarization technique by incorporating word bigram probability and word co-occurrence measure. First we implemented a graph based technique to achieve sentence compression and information fusion. In the second step, we use hand-crafted rule based syntactic constraint to prune our compressed sentences. Finally we use probabilistic measure while exploiting word co-occurrence within a sentence to obtain our summaries. The system can generate summaries for any user-defined compression rate.

KEYWORDS

Text Summarization, Sentence Compression, Hand-crafted Rules, Word Co-occurrence, Word Graph

1. INTRODUCTION

Generating summaries automatically from text document(s) has remained a tough challenge for a long time. The need for such systems is growing due to the large amount of data available at our disposal as well as the redundancy present in those data. Text summarization effectively curtails all the redundancy from text document(s) and presents the user with an informative output. Text summarization can be classified into two broad categories namely abstractive and extractive summarization. We attempt to reach a compromise between these two extremes and formulate a summary containing elements from both the kinds. Extraction based text summarization extracts relevant objects from the given input document(s), without modifying the objects themselves; it merely copies the information deemed most important by the system to the summary. On the other hand, abstraction based summarization does not rely on simple extraction to generate summary; instead it does so by forming new sentences intuitively from the given document(s). Most of the pre-existing works on summarization primarily focus on extractive summaries using keyphrase extraction, similarity detection, rule based strategies, etc. We initially focus on obtaining the sentence compression using an inherent property of the word graph [1] which allows us to obtain pruned and compressed sentences without implementing any learning algorithm or using any language models. Once we get the set of compressed sentences we further modify them by incorporating semantic and syntactic parameters. It is in this latter phase where we use the word co-occurrence measure. We prefer the graph based approach for sentence compression over the more prevalent use of syntactic parser to obtain grammatical compressions because of two main reasons. Firstly, it inherently provides the compressed output sentence without any exterior operation. Secondly, it is simpler and more efficient to implement as it involves graph traversal as opposed to tree traversal. The fact that the word graph within itself contains our desired result is of significant importance as it is computationally much more efficient to traverse a well connected graph rather than traversing a tree. Once the sentence compression phase is over, the syntactic rules can be implemented.

DOI: 10.5121/ijnlc.2016.5502

Filippova and Strube [2] used hand-written rules for syntactic and semantic modification of a parse tree while some researchers [3] used statistical approaches for the tree modification. We took an intermediary approach where we use some hand-crafted rules which were derived from sentence structure observation and analysis. Thereafter we exploited our corpus based knowledge to improve the performance of our system by imposing some probabilistic constraints on the output generation. The resultant output thus undergoes sentence extraction, compression and abstraction. We finally carried out a linguistic evaluation on the Barzillay and Ellahad [4] dataset where our system-generated summaries outperformed theirs. We also evaluated our system generated summaries by comparing them against the Opinosis gold summaries and our results on the Opinosis dataset are comparable with the Opinosis system [5] and significantly better than the MEAD system [6].

2. RELATED WORK

Many text-to-text generation procedures [7, 8] involving sentence compression and abstraction operate on relatively short input file ranging from one sentence to a few paragraphs. Barzilay and McKeown [9] used a sentence fusion technique where they merged phrases based on content relevancy; they also developed the news summarizing application NewsBlaster [10] using a similar concept. Many researchers [11, 12] used machine learning based approaches such as feature extraction to perform summarization tasks and inducing preference among content planning using annotated corpus for training.

Among the graph based methods operating at a sentence level, Radev et al. [13] uses the LexRank algorithm to obtain the centrality scores based on intra-sentence cosine similarity metric whereas Li et al. [14] used a co-ranking based graph algorithm to calculate the relative effect and importance of different sentences. Word level graph compression which was incorporated using a word graph [1] technique was later extended by introducing a key-phrase extraction methodology [15]. Among unsupervised and domain independent methods, Ganesan et al. [5] proposed a graph based abstractive summarization system called Opinosis in which the original input text is represented by a textual graph and the Opinosis system consequently explores this graph and scores the various sub-paths in the graph to generate the candidate summaries. A recent work by Bing et al. [16] works purely on a syntactic front by merging important phrases of maximum salience. Other sentence compression techniques used to incorporate grammatical compression and syntactic modifications include work by [17] who used lexicalized markov grammar for sentence compression whereas [18] created summaries combining dependency subtrees using distributional semantics. Works that involved word co-occurrence information and other statistical measures include Matsuo and Ishizuka [19]. They used keyword extraction for summarization by exploiting statistical information.

3. SYSTEM FRAMEWORK

Our system mainly comprises three major modules which are

1. Sentence Compression
2. Syntactic Modification
3. Probabilistic Modification

Using clustering, we first generate a set of similar sentences from document(s). These sentences serve as the input for word graph generation phase. Once we complete the first phase, we particularly focus on PP attachment handling which forms a major challenge in any summarization task. After we obtain a set of compressed sentences, we create constituency parse tree from those sentences. Then we impose our hand-crafted rules on the sentences to see if we

can prune any parts of the sentences especially any redundant Prepositional Phrase (PP) attachments. Here we impose probabilistic constraints upon selective sections of the sentence to check if they are grammatically and syntactically meaningful. Once all the stages are complete we obtain our output summaries. The underlying process is shown in a rudimentary fashion in Figure 1. The details of each of the stages are discussed in details in the subsequent sections.

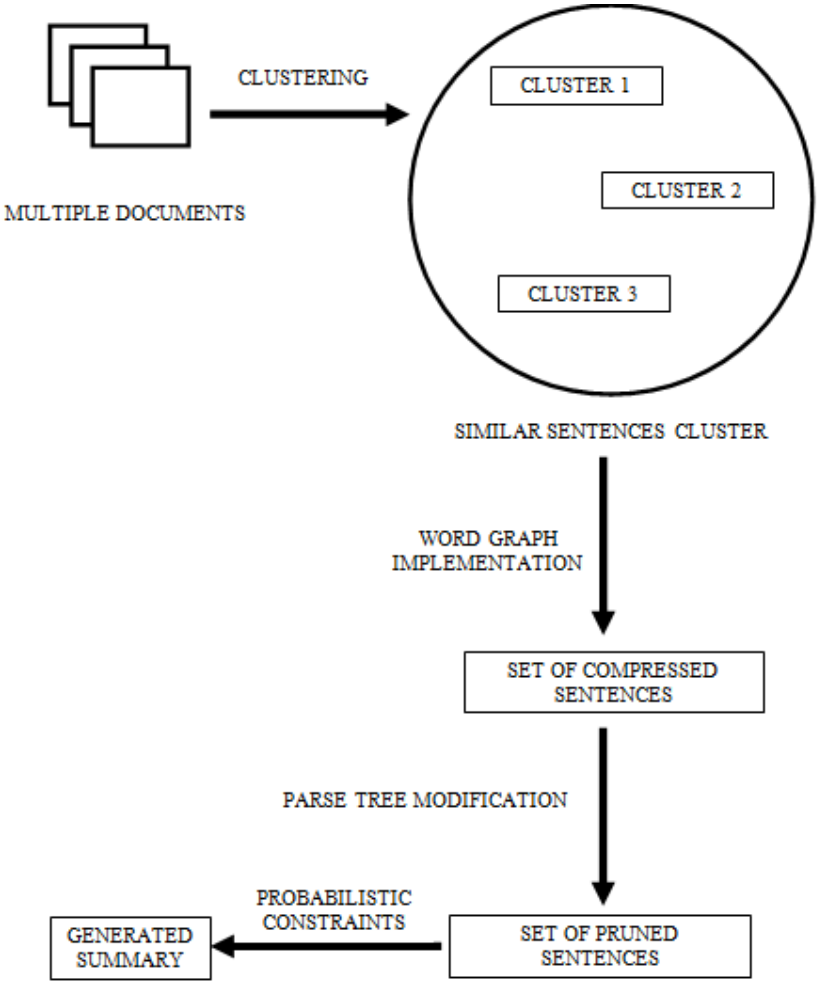


Figure 1: Summarization System Framework

4. SENTENCE COMPRESSION

Sentence compression can be defined as a method for obtaining shorter and more precise sentences from a group of similar sentences while maintaining a syntactic and semantic structure such that the result is grammatically coherent and informatively non-redundant. The sentence compression strategy requires only a POS tagger and list of stop words to work. The NLTK suite [20] was used for the POS tagging as well as for obtaining the stop words. The word graph which represents the cluster of similar sentences operates on the assumption that redundancy in a given set of similar sentences is enough to generate informative texts comprising the important terms because presence of redundancy will ensure that spurious words which do not have much

association with other words will get filtered out since the weighting function is designed in such a way that redundant nodes are assigned less importance.

4.1. Word Graph Construction

Given a set of similar sentences, a word graph is a weighted directed graph where the nodes represent the words of a sentence while the edges represent the connectivity between two adjacent words. The set of similar sentences which serve as the input for the word graph are obtained by using sentence level hierarchical clustering on the input document(s). The steps for a word graph generation are as follows.

a. Starting with the first sentence, the words are added as nodes to the graph one by one.
b. With the addition of every new node, a directed edge from its previous node to the new node is created.
c. With every new node, the connecting directed edges acquire an edge weight of 1.
d. If such a word is encountered which has its equivalent node with a same lowercased form and POS tag then no new node is generated rather the word is mapped onto that existing node.
e. Edge weights are incremented by 1 for the inclusion of an equivalent word node.

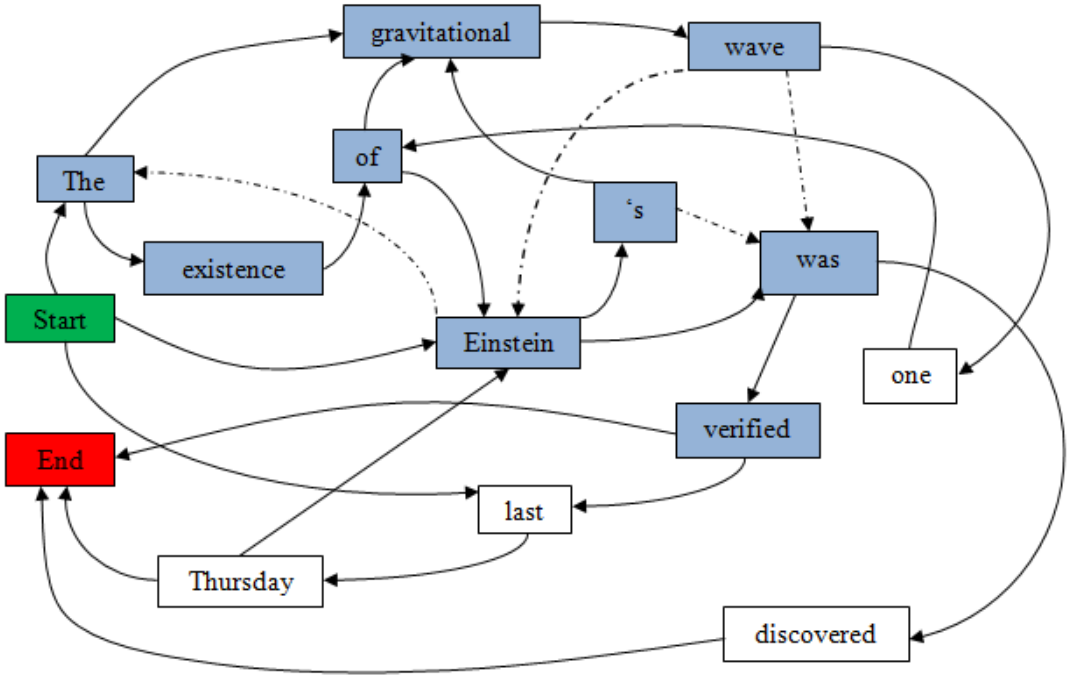


Figure 2: Word Graph generated from sentences (1-4) and a possible compression path

4.2. Illustration of Word Graph

Let us consider the four sentences 1–4 to help understand the workings of a word graph. Edge weights are omitted and the italicized fragments in the sentences are replaced by dotted lines in the figure for the ease of understanding.

The word graph generated from the 4 sentences is shown in Figure 2.

1. The gravitational wave, one of Einstein's *predictions* was verified last Thursday
2. Einstein *predicted* the existence of gravitational wave *and it* was verified
3. Last Thursday, Einstein's gravitational wave was discovered
4. The existence of gravitational wave *predicted by* Einstein was verified

4.3. Compressed Path

While calculating for the compressed path, we set the minimum length of 8 words and the requirement of presence of a verb node for the formation of a sentence. We choose 8 words as any less might lead to incomplete sentences as observed from our dataset. Some of the desirable traits of the compressed path are as follows.

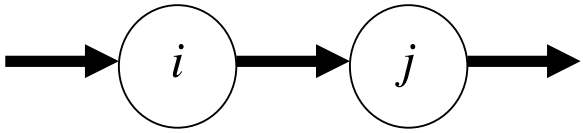
a. Informative nodes
b. Salient nodes
c. Proper order of appearance

The initial Association Function is defined as.

$$AF(e_{ij}) = \frac{W(e_{ij})}{in(i) + out(j)}$$

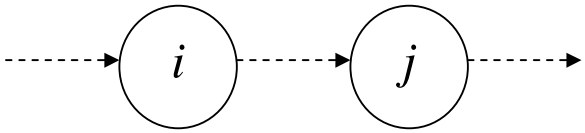
where $W(e_{ij})$ gives the initial weights between two adjacent word (nodes) i and j after the completion of the word graph, $in(i)$ represents the in-degree of the node i whereas $out(j)$ represents the out-degree of the node j. The Association Function strives to minimize the weight between relevant nodes; hence the optimal path will be one with the minimum total path weights. The Association Function maintains the inclusion of strong grammatical compression links, i.e., it favors links between words that occur significantly in an order. It also favors salient words which might have low initial edge weights i.e. $W(e_{ij})$, but have a strong association with adjacent words and removes trivial and common word pairs which might have higher edge weights among themselves but low association and connectivity with other words in its vicinity. This can be illustrated by using 4 different cases. The 4 figures below shows 4 unique word pairing scenario where the incoming arrow on i gives its $in(i)$, the outgoing arrow from j gives the $out(j)$ whereas the arrow between i and j gives $W(e_{ij})$. The bold arrow indicates a heavier edge weight where as the dotted ones signify lesser edge weights.

Case 1:



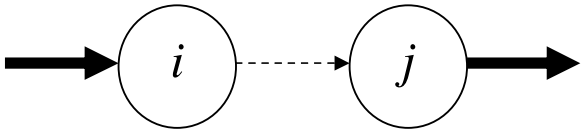
This case shows a scenario where the initial weights in between the two nodes (words) as well as the in-degrees and out-degrees of the respective nodes i and j are high. This shows a high associatively and connectivity all around. Thus, Equation (1) will reduce its original edge weight $W(e_{ij})$ since the larger denominator will always reduce it.

Case 2:



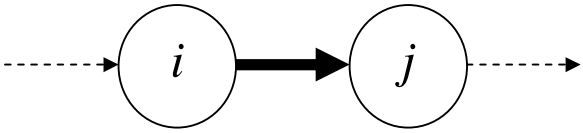
The second case shows the opposite of the previous one where inter-connectivity among the two words as well as their individual in-degree and out-degree are also low. In this case, the weight minimization will be to a much lesser degree hence such paths are less likely to be chosen as optimal paths.

Case 3:



The two cases discussed above are quite simple in principle. The third scenario shows the presence of a salient word, an infrequent word having high connectivity with other words in the vicinity. We mentioned earlier that such word pairs will get higher priority and hence the Equation (1) will minimize the initial edge weight since the denominator of the function will always tend to be greater than the numerator. The total of in-degree and out-degree will always be greater than the edge weight between the words. Thus it is more likely that such word pairs will come up as a part of the optimal compressed path.

Case 4:



The final case shows the presence of a trivial word pair as it can be seen that despite having high initial edge weights their respective in-degrees and out-degrees are much lesser. We assign lower priority to such word pairs. The Equation (1) is less likely to be minimized since the numerator will remain greater.

We redefine our weighing function again using the final Compression Function which is defined as

$$CF(e_{ij}) = \frac{AF(e_{ij})}{freq(i) \times freq(j)}$$

The above function ensures that our compressed path passes through the nodes with highest traffic or frequency as it reduces edge weights among important nodes, i.e. edges with greater frequencies. Let, *freq*(*i*) be the total number of in-degree and out-degree of node *i* while *freq(j)* is the total number of in-degree and out-degree of node *j*. There is also another implied advantage of using this procedure which is Prepositional Phrase (PP) reductions. Pruning of PP attachments is an important aspect in summarizing sentences as the attachments often do not contain any relevant information, can be ambiguous at times and can be redundant as well. With our weight function we try to ensure that such spurious additions get clipped due to their lack of coherent order and low frequency. It is to be noted that the PP reduction method works only if such an attachment exists at the start or at the end of sentence and provided there is not enough high frequency nodes present in the vicinity. The word graph does not actively prunes the PP attachments; rather it is a structural and weighing function advantage. The pruning involving the

prepositional phrases is by no means optimal; we have however discussed in the later section how we sought to rectify this using syntactic and statistical constraints on the formation of sentences. Once we calculate all the edge weights involved, we use *K*-shortest path algorithm to find the fifty shortest paths from the start node to the end node in the graph. Sentences containing minimum 8 words and containing at least a verb node are extracted for the summary. We re-rank all those paths by their length normalized path weight. The path with the minimum average edge weight is our compressed path. Our weighting function is such that frequency of nodes as well as association between the nodes directly correlates with a lower edge weight. This is easily understandable from the weight function as the denominator increases if the node has higher frequency. Thus the minimum edge weight paths are favored.

5. SENTENCE GENERATION

To create an informative summary it is always advisable to incorporate some language specific rules while generating the output sentences. If we solely depended on the word graph approach to create our summary it will suffer from grammatical incongruities at worst and redundant word representation at best. Grammatical incongruities may arise from the inclusion of a word pair which might not make any sense within the context but was nevertheless selected due to their high frequency. Redundant word representation although does not make the output summary incoherent it still degrades the overall quality of the summary. Once we have generated the set of compressed sentences using the word graph, we use them as the input for the next module. From the compressed sentences, we create the constituency based parsed tree using the Stanford CoreNLP suite [21]. Observing and analyzing the constituency parse trees, we put some grammatical constraints on the word graph generated compressed sentences to obtain our final set of sentences for the summary.

5.1. Syntactic Modification

After sentence compression has been performed, we can work on the assumption that we have all the relevant information present in a given sentence; therefore, re-ordering and re-arranging in a grammatically coherent manner would not lead to any loss of information. We make a number of changes in the original structure of the parse tree based on its syntax and semantics. The changes are as follows.

1. Chains of conjuncts are split and each of them is attached to its parent node.
2. Synonyms (with the exception of stop words including determiners and common verb words) are merged into a single word.
3. Any prepositional phrase (PP) attachment appearing at the start of a sentence is dropped iff there exists no such production in the form of ***S→NP VP*** rooted under that PP expansion.
4. If two PP attachments are present such that one is rooted at a NP subtree and the other is at a VP node, then the PP attachment under VP is dropped provided

    I. The PP attachment rooted under a VP node is preceded by the PP at NP node in the sentence.
    II. There does not exist any S production rooted under the PP subtree or VP subtree
    III. There does not exist any verb phrase expansion under the PP subtree

5. Basic sentence formation constraint must be maintained i.e. no sentence can have less than 8 words and they must have at least one verb node.

Now we will proceed to show how some of the grammar rules we have outlined work with few examples

Example 1: *in recent times only few students are opting for STEM courses*

Its corresponding parse tree is given in Figure 3. According to rule (3) stated above since a PP attachment exists at the start of the sentence and no production of the form *S→NP VP* lies beneath it, the attachment "*in recent times*" will get pruned.

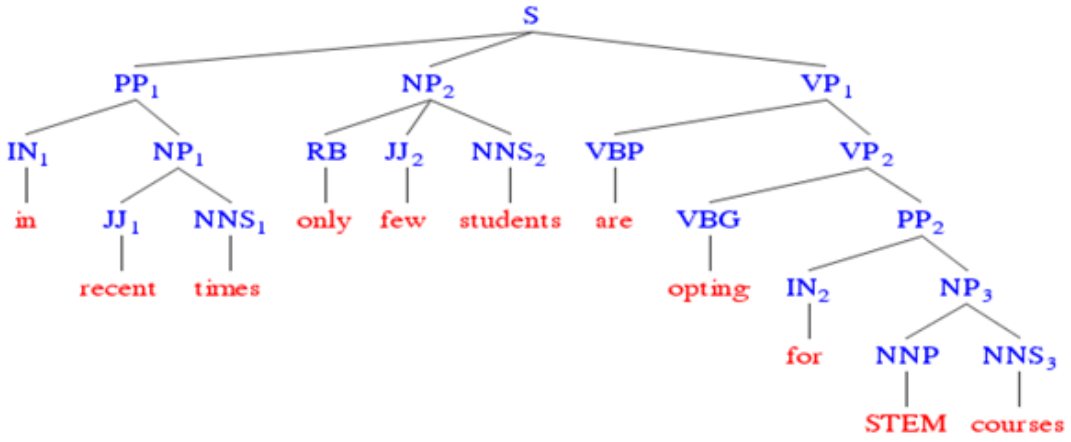


Figure 3: Parse Tree generated for the Example 1

Example 2: *the existence of gravitational wave was announced yesterday in the early morning*
The corresponding parse tree is given below in Figure 4. According to rule (4) described above, the PP attachment "*in the early morning*" is pruned as another PP attachment under a NP node already precedes it

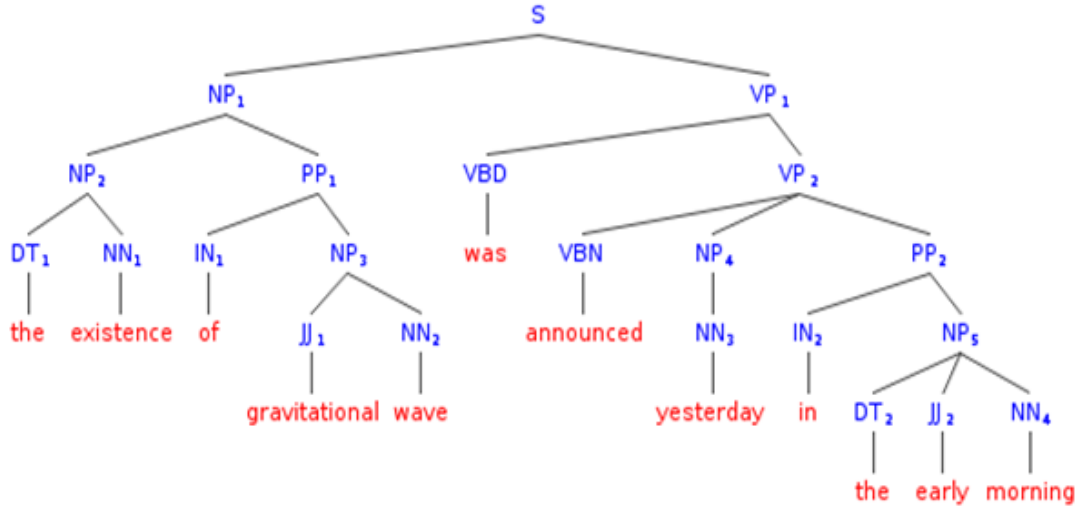


Figure 4: Parse Tree generated for the Example 2

Example 3: *the lecturer in his haste skimmed through the presentation to make his evening appointment*

The corresponding parse tree is given below in Figure 5. Despite having a PP attachment under VP subtree, it cannot be pruned according to rule (4) since there exists a S production beneath the VP subtree. If pruning is not possible (as shown in this example) and there still remain multiple

PP attachments in the sentence, we will use bigram probability and word co-occurrence probability to further modify the sentence which we will show in the next section.

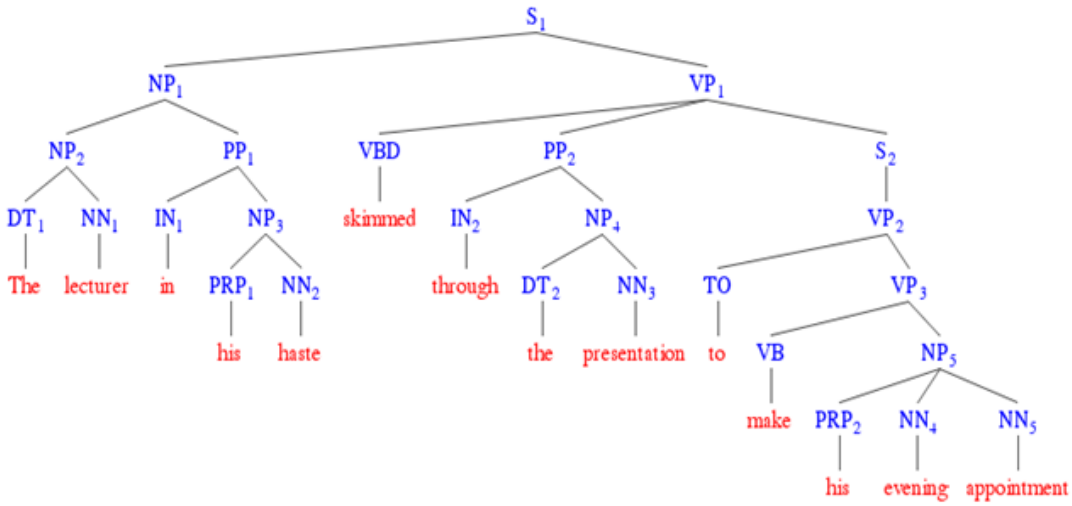


Figure 5: Parse Tree generated for the Example 3

5.2. Probabilistic Modification

Even the most well crafted grammar rules might become too generic to filter out redundancy. Therefore, as a final restriction we introduce a probabilistic constraint parameter to prune edges using conditional bigram probability. It might so happen that PP attachments, not covered under the previously discussed conditions, are still present in the generated sentences. In such cases, bigram probability of the preposition head term at the start of a PP attachment is calculated with respect to its previous term. Here the prepositional head term refers to the preposition node at which a PP subtree is rooted. If the conditional probability of a preposition head term occurring at a latter position down the sentence has a lower probability than that of a preposition term occurring at an earlier position within that sentence then the latter PP attachment is pruned. The condition for this pruning is given as:

$$Pr\,(PP_l \mid prev\ term) < Pr\,(PP_f \mid prev\ term)$$

Where $Pr\,(PP \mid prev\ term)$ is the bigram conditional probability of the preposition term at the start of a prepositional phrase given the previous word. PP_f and PP_l are the two preposition terms among which we make our comparison where PP_f refers to the preposition term which occurs earlier in the sentence and PP_l refers to the prepositional term occurring later in the sentence. This pruning follows the obvious logic that if such PP attachments are present in the sentence at such great depth it might be conveying important information. However, at the same time, given their position in the sentence we know that such attachments have persisted mainly because of their structure which allowed it to be retained so far into the process. Thus we reach a compromise by checking their probabilistic significance from a commonly used, versatile, annotated corpus to see how they fare.

The second part of probabilistic comparison involves word co-occurrence measure. If the inequality presented above doesn't hold i.e. if the bigram conditional probability of PP_l is greater than PP_f then we measure the word co-occurrence count. If the word co-occurrence probability of the former PP head term is lesser than the co-occurrence probability of the next term that comes after the PP subtree, then the PP subtree is pruned and the next term is annexed. The example below illustrates how it works. *The teacher in his haste skimmed through the presentation to make his evening appointment* Here "*in*" being the first PP head while "*through*" being the

second PP head. So if the bigram conditional probability of PP_f i.e. "*in*" is less than that of PP_l i.e. "*through*", we can prune the former PP subtree rooted at "*in*" if and only if the co-occurrence probability of "*in*" is lesser than the co-occurrence probability of the next word present after the PP subtree. In our example, "*in his haste*" is rooted under the PP subtree and "*skimmed*" comes next in line to be annexed after "*teacher*". Thus if co-occurrence probability of "*in*" in the context window of "*teacher in his*" is less than the co-occurrence probability of "*skimmed*" in the context window of "*teacher skimmed through*" we drop the PP subtree. We use this technique under the assumption that if co-occurrence probability of the new word that is to be annexed is higher than the word already present in the original formation there's a strong bias in favor of the new word to form a grammatically and syntactically coherent sentence. Since the original formation was already syntactically valid the change based upon co-occurrence must be at the least equally valid. Here co-occurrence measure is used as a metric for syntactic and grammatical validity. We use a co-occurrence window of 1 word i.e. previous one word and next one word. We calculate the co-occurrence probability after obtaining the co-occurrence counts from the word co-occurrence matrix developed from the corpus. The co-occurrence probability is given by:

$$Pr\,(t) = (1\,/\,N)\,\Sigma\; counts\,(c,\,t)$$

Where $Pr\,(t)$ is the target word co-occurrence probability, N being the total word count in the corpus and $counts\,(c,\,t)$ gives the word co-occurrence counts of the target word within a window size of 1 word on its left and right.

We use the Corpus of Contemporary American English [22] for calculating the bigram probabilities and the word co-occurrence counts. The corpus is composed of 520 million words from more than 160,000 texts. The corpus is evenly divided between the five genres: spoken, fiction, popular magazines, newspapers, and academic journals. It is to be noted that all the above syntactic and statistical rules are not meant for universal application. Our summarization system mostly deals with formalized data as its input and as such all those data follow a particular grammatical structure. By studying and testing all the rules on our working data we have come to the conclusion of imposing them as they fit our work the best. For a different type of dataset such as unstructured data like social media feeds, the application of the same rules might not yield the desired results. However for formalized dataset like news articles, academic journals, review opinions, academic texts etc. our approach will work efficiently.

6. EVALUATION

It is difficult to evaluate abstractive summarization tasks due to the lack of any natural baseline for comparison. For automatic evaluation, we used the ROUGE 2.0 [23] metric to quantitatively assess our system generated summaries against human generated gold standard summaries. Here we used the Opinosis dataset as the gold standard for comparing our system against the MEAD and Opinosis systems. The Opinosis dataset comprises sentences extracted from reviews on any particular topic. There are 51 such topics and the reviews on each topic provide the input text which needs to be summarized. The dataset also contains human generated summaries for the said topics which we use as the gold standard model summaries. MEAD [6] is a state of the art extractive summarization tool which uses three features namely centroid, text position and sentence length to score and rank sentences to generate summaries. We compared our system with Opinosis because of its operational proximity to our proposed system. It uses a mix of graph based approach and statistical approach which is quite similar to ours. Since most of the gold standard summaries ranged from 2 sentences to 4 sentences, while evaluating with the Opinosis gold standard dataset we generated our summaries at a compression rate of 95% which corresponded well length wise with the gold standard summaries in the Opinosis dataset. The

comparative evaluation of our system, the Opinosis system and MEAD on the Opinosis test set are presented in Table 1

ROUGE-1	Average Precision	Average Recall	Average F-Measure
Gold Standard	1	1	1
Our System	0.23	0.52	0.32
Opinosis	0.42	0.28	0.33
MEAD	0.10	0.49	0.16

Table 1: Evaluation with Rouge-1 against Opinosis Gold Standard

We also instructed human reviewers to generate summaries of the input documents present in the Opinosis test set. Two sets of summaries were generated for each topic, one at a compression rate of 85% and the other at 95%. Keeping in mind the constraint of our system, the reviewers were told of compose each sentences having minimum of 8 words and one VB node. We used the Rouge-1 and Rouge-2 metric to make a comparative analysis of how our system fares against our human generated summaries. The results are presented in Table 2.

Metric	Compression Rate	Average Precision	Average Recall	Average F-Measure
Rouge-1	95%	0.82	0.38	0.52
	85%	0.85	0.63	0.72
Rouge-2	95%	0.76	0.31	0.44
	85%	0.81	0.60	0.69

Table 2: Evaluation with Rouge-1, 2 against Human Summaries

After the quantitative evaluation of our system, we proceed towards the qualitative evaluation. The linguistic quality of the summaries are assessed using the five linguistic quality questions as in Bing et al. [16] which includes (Q1) grammaticality, (Q2) non-redundancy, (Q3) referential clarity, (Q4) focus and (Q5) coherence. A 5-point Likert scale is used where 1 represents very poor and 5 being very good. We used the test set provided by Barzilay and Elhadad (B&E) to make a comparative analysis against the B&E system [4]. For our qualitative comparison we chose this system as the baseline because of its extractive nature which makes it inherently more grammatically sound compared to abstractive systems. Additionally, the concept of lexical chains used by the B&E system exploits the co-dependency among its words to form sentences which is comparable to our method of using word co-occurrence to form new sentences. Summaries produced by our system and the B&E system on each of the 24 test set topics were blindly evaluated by 3 assessors and the result of this evaluation is presented in Table 3.

Linguistic Parameter	Q1	Q2	Q3	Q4	Q5	Avg
Our System	4.05	3.91	4.02	4.5	3.63	4.02
B&E	4.31	3.81	3.50	4.0	2.94	3.71

Table 3: Evaluation of Linguistic Quality at 85% Compression Rates

Furthermore, for a better understanding of the comparison among the systems we use the Rouge-1 metric to calculate the recall, precision and F-measures of our system generated summaries on the B&E dataset and compare it against the B&E summaries. Similar to the qualitative evaluation, our system outperforms the B&E system in the quantitative evaluation as well. The results of that comparison are presented in Table 4.

ROUGE-1	Average Precision	Average Recall	Average F-Measure
Our System	0.61	0.69	0.65
B&E	0.47	0.62	0.53

Table 4: Evaluation with Rouge-1 against B&E dataset at 85% Compression Rates

We also do a readability test to judge whether our system generated summaries are readable enough. For that we use the [5] formula for measuring readability and subsequently compare our results with theirs. Suppose there are N sentences in our system generated summary and M sentences in the corresponding human summary. We mix all these sentences and then ask a human assessor to pick at most N number of sentences that they deem as least readable.

$$\text{Readability} = 1 - (\#\text{CorrectPick} / N)$$

The human assessors were instructed to pick the least readable sentences from each of the 51 summaries for the Opinosis dataset. Our system at an 85% compression rate generated 218 sentences in total from all of our summaries combined and the human assessors picked 50 sentences which they deemed least readable. Thus we get a readability score of 0.77 which outperforms $\text{Opinosis}_{\text{best}}$ score of 0.67. We also tested the readability scores of our system for higher compression rate summaries; however, the score gradually degraded as compression rate was increased which is expected. Thus the summaries obtained at 85% compression rate can be considered the most optimal performance by our system.

7. RESULTS AND ANALYSIS

Table 1 presents the Rouge-1 scores obtained against the Opinosis gold standard. Evaluation results on machine generated summaries from the Opinosis and MEAD systems are also presented in Table 1 to provide a comparative understanding of how our system fares with respect to Opinosis and MEAD. The proposed system outperformed the MEAD baseline by a significant margin, a 100% improvement in F-measure. Additionally, paired t-test (with $p = 0.01$) comparing our system with MEAD baseline against Rouge-1 metric shows our improvement over the baseline to be of statistical significance. Overall the Opinosis system performs marginally better than our system. However it should be noted that the Opinosis summaries ranged from 2 sentences to 4 sentences whereas our system generate summaries at desired compression rate and as we discussed earlier we had to set compression rate at 95%. As we have shown in section 5, manual evaluation performed by reviewers with respect to linguistic quality and readability show that our system produces the most optimal summaries at around 85% compression rate, thus at high compression rates of 95%, a slight dip in the F-measure in comparison to Opinosis' best performance is within acceptable limits. Therefore, overall the performance of our system can be considered comparable to that of the Opinosis system. Our system fails to produce better results at higher compression rates primarily because of a restriction imposed during the initial sentence compression stage i.e. maintaining a minimum of 8 words in every generated sentence. It fails to produce informative summaries at higher compression where shorter sentences with more

selective set of words might have been able to convey the information more succinctly. We focus more on the content and structure of the generated sentences over individual isolated words. Therefore, at higher compression rates, our system produces fewer sentences while maintaining the minimum sentence length as opposed to generating greater number of shorter sentences. This is a limitation of the proposed system which can be addressed by incorporating keyphrase extraction methods, supervised content selection technique, tuning of the optimal sentence compression rate, sentence length, etc. Furthermore if we look at the ROUGE scores presented in Table 1, it can be observed that precision of our system is quite low compared to that of the Opinosis system. This is because our system's output were longer sentences with some spurious terms as opposed to theirs which were shorter and more relevant. Thus the fraction of relevant terms among the retrieved instances was lower for our system. In Table 2, the Rouge-1 and Rouge-2 evaluation of the system against human summaries for two different compression rates are presented. The problem that was encountered during the Opinosis comparison doesn't happen in this case since the human reviewers were already instructed to maintain the 8 word sentence structure in the first place. Even the F-measure at 95% compression rate is significantly better than the one obtained by comparing against Opinosis gold standard summaries at the same compression rate. At 85% compression rate, the system reaches its peak performance as evident from both the quantitative and qualitative evaluation.

The Rouge results evaluate our summaries at the word level whereas readability evaluates them at the sentence level. From our readability scores of 77% we can say that most of the system generated summaries are in fact indistinguishable from the human summaries. Table 3 presents the linguistic quality evaluation. The B&E is an extraction-based method that creates sentences using lexical chains [4]; hence it achieves a slight higher score in Q1 grammaticality, while our approach generates some new sentences with grammatical mistakes, a common hindrance for abstractive methods. For Q5 coherence, our score is higher than B&E, which reveals that our summary sentences are relatively more structured and meaningful. The score of Q3 referential clarity shows that the referential relations even among newly generated sentences are better than an extractive system. On average, scores of our system surpasses the baseline system. Thus we use two dataset i.e. Opinosis and B&E to independently evaluate our system and the results of our system outperform both the MEAD and B&E baseline system while it is comparable to the Opinosis system.

8. CONCLUSIONS

We presented in this paper a compression based text summarization method which uses a graph based technique to achieve sentence compression. It does not rely on any supervised technique or similarity measure for content selection; instead, it uses an efficient weighting function to obtain the important and salient nodes from a cluster of related sentences obtained from multiple documents. We then apply some syntactical rules to ensure that any more redundancies, if present, are eliminated. As a final step, we introduced a Probabilistic constraint imposing a stricter selection in grammatical structure. Sentence compression using word graph is performed only once for the entire dataset. However the syntactic and statistical constraints can be re-iterated until a desired compression rate is achieved by the output summary. Lastly, we presented in details the evaluation of our system and explained its strengths and shortcomings. Our approach takes a compromise between abstraction and extraction as we incorporate features from both the types. However, it must be noted that there is not any hard line where one stops and the other begins. The extraction and abstraction are intertwined and occur at every step of the process.

9. FUTURE WORK

Abstractive text summarization has a lot of potential and new methods and techniques are being applied with promising results. As for our approach, this too has room for improvement. [1] used her own version of scoring mechanism, while we used our own scoring method; it is always possible to come up with a better scoring function which is more inclusive of all the information and salience present within the sentence context. As for the syntactic and semantic components of an abstractive summarization system, it is not very practical to always depend upon hand-crafted rules as exceptions are always there and can quite significantly degrade the system performance. Reliance on a more statistical approach can also make the process dependent upon an external source for support, be it a training corpus or machine learning approaches which might not always work in conjuncture with the input documents. The innate abstraction present in summaries makes the task of automatic abstractive text summarization considerably challenging to find a foolproof solution. In future we wish to improve our scoring algorithm which will lead to more informative summaries and also to incorporate more fine-grained phrase merging to enhance the grammatical texture of the output summaries, which in turn will help the system to perform better at higher compression rates.

REFERENCES

[1] Katja Filippova, 2010. Multi-sentence compression: Finding shortest paths in word graphs. In Proceedings of the 23rd International Conference on Computational Linguistics, pp 322–330. ACL

[2] Katja Filippova and Michael Strube, 2008. Sentence fusion via dependency graph compression. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp 177–185. ACL

[3] Daniel Marcu and Kevin Knight, 2000. Statistics based Summarization-step one: Sentence compression. American Association for Artificial Intelligence, Vol 139, pp 91-107. Elsevier

[4] Michael Elhadad and Regina Barzilay, 1999. Using lexical chains for text summarization. Advances in Automatic Text Summarization, pp 111-121. The MIT Press

[5] Kavita Ganesan, ChengXiang Zhai and Jiawei Han, 2010.Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In Proceedings of the 23rd International Conference on Computational Linguistics, pp 340–348, Beijing

[6] Dragomir Radev, Sasha Blair-Goldensohn and Zhu Zhang, 2001. Experiments in single and multidocument summarization using MEAD. In First Document Understanding Conference New Orleans, LA

[7] Hongyan Jing and Kathleen R. McKeown, 2000. Cut and Paste based text summarization. In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference. ACL

[8] James Clark and Mirella Lapata, 2008. Global inference for sentence compression: An integer linear programming approach. Journal of Artificial Intelligence Research, pp 399-429

[9] Kathleen McKeown and Regina Barzilay, 2005. Sentence fusion for multi-document news summarization.Computational Linguistics, Vol 31(3), pp 297–328. ACL

[10] Kathleen R McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, Sergey Sigelman,2002. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In Proceedings of the second international conference on Human Language Technology Research pp 280-285. Morgan Kaufmann Publishers Inc.

[11] Joel L. Neto, Alex A. Freitas, Celso A. A. Kaestner, 2002. Automatic text summarization using a machine learning approach.Advances in Artificial Intelligence.pp 205-215. SBIA

[12] Kathleen McKeown and Min Y Kan, 2002. Corpus trained text generation for summarization. In Proceedings of the Second International Natural Language Generation Conference, pp 1-8

[13] Dragomir R Radev and Gunes Erkan, 2004. LexRank: graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, pp 457-479

[14] Chen Li, Xian Qian, and Yang Liu. 2013. Using supervised bigram-based ILP for extractive summarization. In Proceedings of ACL

[15] Florian Boudin and Morin Emmanuel, 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 298–305. ACL
[16] Lidong Bing, Piji Li, Rebecca J. Passonneau, Wai Lam, Weiwei Guo, Yi Liao, 2015. Abstractive Multi-Document Summarization via Phrase Selection and Merging , pp 1587-1597. ACL
[17] Kathleen R. McKeown and Michel Galley, 2007. Lexicalized Markov grammars for sentence compression. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp 180–187. ACL
[18] Gerald Penn and Jackie C. K. Cheung, 2014. Unsupervised sentence enhancement for automatic summarization. EMNLP pp 775-786. ACL
[19] Yutaka Matsuo and Mitsuru Ishizuka, 2004. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, Vol 13, pp 157-169. World Scientific Publishing Company
[20] Edward Loper, Ewan Klein and Steven Bird, 2009. Natural Language Processing with Python. O'Reilly Media Inc.
[21] Christopher D. Manning, David McClosky, Jenny Finkel, Mihai Surdeanu, John Bauer and Steven J. Bethard, 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp 55-60. ACL
[22] Mark Davies, 2009. The 395+ million word corpus of contemporary American English (1990–2008þ): design, architecture, and linguistic insights. International Journal of Corpus Linguistics, Vol 14, pp 159–190. John Benjamins Publishing Company
[23] Chin-Yew Lin, 2004. Rouge: A package for automatic evaluation of summaries. In Proceedings of the workshop of Text Summarization branches out, Vol 8. ACL-04

Authors

Shouvik Roy completed his Masters in Computer Science and Engineering from Jadavpur University, India in 2016. His research topics and area of interests include Natural Language Processing, Text Summarization, Natural Language Generation and Machine Learning.