Int J Parallel Prog (2018) 46:992–1016
https://doi.org/10.1007/s10766-018-0560-3

Abstractive Text Summarization based on Improved Semantic Graph Approach

Atif Khan[1] · **Naomie Salim**[2] · **Haleem Farman**[1] · **Murad Khan**[3] · **Bilal Jan**[4] · **Awais Ahmad**[5] · **Imran Ahmed**[6] · **Anand Paul**[7]

Received: 15 July 2017 / Accepted: 24 January 2018 / Published online: 2 February 2018

Abstract The goal of abstractive summarization of multi-documents is to automatically produce a condensed version of the document text and maintain the significant information. Most of the graph-based extractive methods represent sentence as bag of words and utilize content similarity measure, which might fail to detect semantically equivalent redundant sentences. On other hand, graph based abstractive method depends on domain expert to build a semantic graph from manually created ontology,

✉ Atif Khan
atifkhan@icp.edu.pk

Naomie Salim
drnaomiesalim@gmail.com

Haleem Farman
haleem.farman@icp.edu.pk

Murad Khan
murad.csit@suit.edu.pk

Bilal Jan
bilal.jan@fu.edu.pk

Awais Ahmad
aahmad.marwat@gmail.com

Imran Ahmed
imran.ahmed@imsciences.edu.pk

Anand Paul
paul.editor@gmail.com

[1] Department of Computer Science, Islamia College, Peshawar, Pakistan

[2] Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia

[3] Department of Computer and IT, Sarhad University of Science and IT, Peshawar, Pakistan

[4] Department of Computer Science, FATA University, Dara Adam Khel, FR Kohat, Pakistan

which requires time and effort. This work presents a semantic graph approach with improved ranking algorithm for abstractive summarization of multi-documents. The semantic graph is built from the source documents in a manner that the graph nodes denote the predicate argument structures (PASs)—the semantic structure of sentence, which is automatically identified by using semantic role labeling; while graph edges represent similarity weight, which is computed from PASs semantic similarity. In order to reflect the impact of both document and document set on PASs, the edge of semantic graph is further augmented with PAS-to-document and PAS-to-document set relationships. The important graph nodes (PASs) are ranked using the improved graph ranking algorithm. The redundant PASs are reduced by using maximal marginal relevance for re-ranking the PASs and finally summary sentences are generated from the top ranked PASs using language generation. Experiment of this research is accomplished using DUC-2002, a standard dataset for document summarization. Experimental findings signify that the proposed approach shows superior performance than other summarization approaches.

Keywords Multi-document abstractive summarization · Semantic graph · Semantic role labeling (SRL) · Semantic similarity measure · Graph based ranking algorithm

1 Introduction

In the current era of information overload, multi-document summarization (MDS) is primary tool that generates a concise summary while maintaining the relevant content of source documents [1,2]. Generally, two approaches are used for MDS i.e. extractive and abstractive. Extractive summarization extracts salient sentences from the text documents and merge them to create a summary without altering the source text. However, abstractive summarization usually employ semantic methods and language generation techniques [3,4] to create a concise summary that is closer to way humans create.

The first attempt on automatic summarization was performed in the late 1950 [5]. The approach in [5] utilizes term frequencies to assess the sentence importance i.e. sentences are included in the summary if they contain high frequent terms.

Recently, MDS is gaining more attention in research community. Most research studies in multi-document summarization have paid attention to extractive summarization, which make summary by selecting salient sentences from the documents [6]. Statistical methods are frequently employed to find keywords and phrases [7]. Discourse structures also assist in identifying the most significant sentences in the document [7]. A range of machine learning (ML) techniques are also utilized to extract

[5] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Republic of Korea

[6] Institute of Management Science, Peshawar, Pakistan

[7] School of Computer Science and Engineering, Kyugpook National University, Daegu, Republic of Korea

features for relevant sentences using training corpus [8,9]. Various graph-based methods [10–13] have also been explored for extractive summarization of multi-documents. These methods employ PageRank algorithm and its variations for computing the relative importance of sentences. A few research studies have considered MDAS in academia. Abstractive summarization methods are generally grouped into two categories: Linguistic (Syntactic) and semantic based approaches. Syntactic approaches for abstractive summaries include tree based method [2,14], lead and body phrase method [15] and information item based method [16]. However, semantic based approaches mainly use template based methods [17,18] and ontology based methods [19–21].

Multi-document summary normally presents a brief topic description for collection of documents on same topic and assist the users to quickly scan many documents. A definite problem for MDS is that there is certain overlapping information in different documents about same topic. Hence, efficient summarization methods that combine similar information content across different documents are required [2]. In this connection, numerous methods have been devised but suffer from some limitations. In particular, the above mentioned graph-based models presented for multi-document extractive summarization (MDES) represent sentence as vector of words without comprehending its meaning. These models make use of content similarity to find sentence similarities, which may not be able to detect semantically equivalent redundant sentences, and hence will lead to a poor final summary. On other hand, the graph-based abstractive summarization approach [21] depends on humans and is restricted to single domain i.e. not applicable to other domains. Based on our literature knowledge, we know that semantic graph-based method has not been considered for MDAS. Therefore, this study introduces a semantic graph-based method for MDAS, which attempts to overcome the disadvantages of existing graph-based approaches. The contributions of this research are highlighted as given:

- Introduce a semantic graph approach for MDAS.
- Improve graph-based ranking algorithm by taking into consideration PASs semantic similarity, and two kinds of PAS relationships.
- Integration of semantic similarity in the graph-based approach to determine semantic relationship between PASs, which will assist in detecting redundancy. PASs are assumed to be redundant if their similarity threshold is greater than 0.5. In other words, no link is established between PASs in the semantic graph whose similarity exceed 0.5.
- Propose a voting method for arrangement of sentences in the multi-document summary.

The remainder of paper is arranged as follows: Sect. 2 describes the related work to this research. Section 3 illustrates the proposed approach. Section 4 presents the evaluation results and discussion. Finally, we end with conclusion and future work in Sect. 5.

2 Related Work

In this section, at first we demonstrate the previous approaches introduced for MDAS, then we discuss graph-based approaches proposed for multi-document extractive summarization and single document abstractive summarization. Finally, we concisely illustrate our proposed graph-based approach for MDAS. A few researchers have strived to create abstractive summaries using a variety of methods, which can be organized into two classes: Syntactic(or Linguistic) and Semantic approach. All linguistic based approaches [2,14–16] proposed for abstractive summarization utilize syntactic parser to represent and analyze the text. The notable disadvantage of these studies is the non-existence of semantic representation of source document text. It is extremely important to represent the text semantically, as profound semantic text analysis is carried out in abstractive summarization. However, different semantic approaches have been investigated for MDAS and are demonstrated as follows.

GISTEXTER is MDS system presented by [17], which employs template based method to generate abstractive summaries from numerous news documents. A main drawback of this method was that extraction rules or linguistic patterns were manually generated, which require more effort and time. A fuzzy ontology approach [19] is presented for summarization of Chinese news, which models uncertain information to describe the domain knowledge in a better way. In this approach, the Chinese dictionary and domain ontology need to be defined precisely by a human expert, which is a tedious task. A framework presented by [20] generates abstractive summary from the semantic model, representing a multimodal document. The knowledge represented by concepts of ontology is utilized to build the semantic model. The weakness of this framework is that it depends on human expert to construct domain ontology, and does not apply to other domains. The methodology presented by [18] produces well written and concise abstractive summaries from the groups of news articles on similar topic. The limitation of the methodology was that generation patterns and information extraction (IE) rules were manually written, which requires effort and time.

Most recently, different graph-based models have gained more consideration and successfully attempted for MDS. These models employ PageRank algorithm [22] and its variations to assign ranks to sentences or passages. Reference [23] presented a connectivity model based on graph, which assumes that nodes which are linked to several other nodes are most probably to carry significant information. Lex-PageRank [10] is an approach that employs the concept of eigenvector centrality for determining the importance of sentence. It builds a sentence connectivity matrix and uses algorithm similar to PageRank to determine the important sentences. Another similar algorithm to PageRank is presented by [12], which determine the salience of sentence for document summarization. Reference [24] introduced a graph based approach, which combines text content with surface features, and investigates the features of sub-topics in multi-documents to include them into the graph-based ranking algorithm. An affinity graph-based approach for summarization of multi-documents [13] employs similar algorithm to PageRank, and calculates scores of sentences in the affinity graph based on information richness.

However, all these graph methods discussed so far did not consider predicate argument structure i.e. the semantic structure of sentence. Furthermore, these approaches did not assume semantic relationships existing between sentences while determining the importance score of sentences.

Reference [25] investigated a graph-based document-sensitive method for generic summarization. However, the model lacks semantic relationships between sentences.

A weighted graph model for generic summarization of multi-documents is introduced by [26] that combines sentence ranking and sentence clustering methods. However, this approach also did not take into account semantic relationships between sentences. Reference [27] introduced a graph based method for multi-document summarization of Vietnamese documents. However, semantic similarity methods are not applicable to Vietnamese documents due to the lack of lexical resources such as English WordNet. Reference [28] conducted a series of analytical studies to compare system summaries with human-written summaries on the basis of semantic units (case frames). However, the studies did not present any summarization system.

Reference [29] demonstrated an event graph-based approach for multi-document extractive summarization (MDES), which combines machine learning with hand crafted rules to extract sentence-level event mentions, and employs a supervised model to determine temporal relations between them. However, construction of hand crafted rules for argument extraction is a time consuming task and may limit the approach to a particular domain.

The above graph-based models discussed for MDES consider sentence as vector of words without taking into account its semantic structure. These models determine sentence similarities based on traditional cosine similarity, which may not be able to detect semantically equivalent redundant sentences, and therefore leads to poor final summary with redundancy.

On other hand, a graph based approach [21] has also been attempted for abstractive summarization, which creates semantic graph for document from human built ontology. The approach depends on human experts to a great extent and is restricted to a specific domain. Secondly, the approach did not report any summarization results. Reference [30] introduced framework for abstractive summarization, in which the document text is represented by a set of abstractive meaning representation (AMR) graphs, which are transformed into a text summary graph. Finally, graph-to-graph transformation is made to produce a summary graph, which is used for generation of summary content. The approach introduced by [31] employs integer linear optimization to pick and merge fine-grained syntactic units such as noun/verb phrases to construct new summary sentences. Reference [32] presented an approach that employed the idea of Basic Semantic Unit (BSU) to describe the semantics of an event or action. The approach captures semantic information of the text documents by defining a semantic link network, which takes into account BSUs. The sentences produced by the semantic link network constitute the structure of the summary. Reference [33] introduced an abstractive approach based on generalization and concepts fusion, i.e. different concepts occurring in a sentence can be substituted by a concept which cover the semantics of all concepts. The approach uses heuristic-based and Machine Learning-based (SVM) models to reduce the space of generalizations and finally a compressed sentence is generated from the best generalization version found. Reference [34] presented a comprehensive survey on recent automatic summarization techniques for extraction as well as abstraction. Various research studies have also explored applications of data mining in different fields such as the Internet of Things and smart cultural heritage spaces [35–39].

Based on our literature knowledge, we came to know that semantic graph-based abstractive approach has not been considered for MDS. The proposed approach differs from previous graph based approaches in the following manner: Existing graph based approaches are presented either for multi-document extractive summarization (MDES) or single document abstractive summarization (SDAS); while our work propose a semantic graph-based method for MDAS. Graph based approach for SDAS relies on human experts to make domain ontology (from which semantic graph is built). Our approach, on the other hand, builds semantic graph by integrating SRL with graph, which can be applied on any domain without relying on human expert. Next, considering graph based approaches for MDES, most of them consider sentence as vector of words without taking into account its meaning. Our work determine PAS from each sentence in order to capture the semantics of sentence (e.g. who did what to whom, when and how). Furthermore, previous graph based approaches for MDES determine sentence relationships based on content similarity measure (i.e. cosine similarity measure) and ignore semantic similarity measure. However, we incorporate semantic similarity between PASs, and two kinds of PAS relationships into the ranking algorithm. The next section discusses the proposed approach in detail.

3 Overview of Approach

The architecture of our graph-based approach is given in Fig. 1. We begin by segmenting the document collection into sentences. Each sentence is given a key based on its document number and location number. Then, we employ SENNA semantic role parser [40,41] to determine PAS from the sentence collection. Next, we build a similarity matrix from the pair wise semantic similarities of predicate argument structures. A semantic graph (an undirected weighted graph) is created from the similarity matrix (as discussed in Sect. 3.2) in a manner that if similarity weight $sim\left(p_i, p_j\right)$ between PASs (vertices) p_i and p_j is greater than zero, then link is set up between them; otherwise a link is not created. This study defines a semantic similarity threshold i.e. $0 < \beta \leq 0.5$ based on empirical analysis [12]. Consequently, a link is created between PASs (vertices), if semantic similarity is within the range; otherwise no link is set up if similarity is outside the range.

In order to reveal the influence of both document and document set on PASs, the edge of semantic graph is augmented with PAS-to-document and PAS-to-document set relationships, whereas the graph edge represents semantic similarity weight between PASs. The importance (salience) score of vertices (PASs) is computed using improved weighted graph ranking algorithm (i.e. we incorporated semantic similarity between PASs, and two kinds of PAS relationships into the graph-based ranking algorithm), as discussed in Sect. 3.2. The ranking algorithm recursively computes the importance scores of graph vertices till the convergence is attained.

As the ranking algorithm converges, the importance/rank scores for the vertices are arranged in reverse order. However, the ranking algorithm may assign same score to the similar predicate argument structures representing the same concept.

Thus, in order to reduce redundancy, the proposed approach utilizes maximal marginal relevance (MMR) for re-ranking the PASs; and then the top ranked vertices (PASs) are chosen based on compression rate of summary and fed to summary generation phase. At last, the Simple NLG engine [42] is used to produce summary

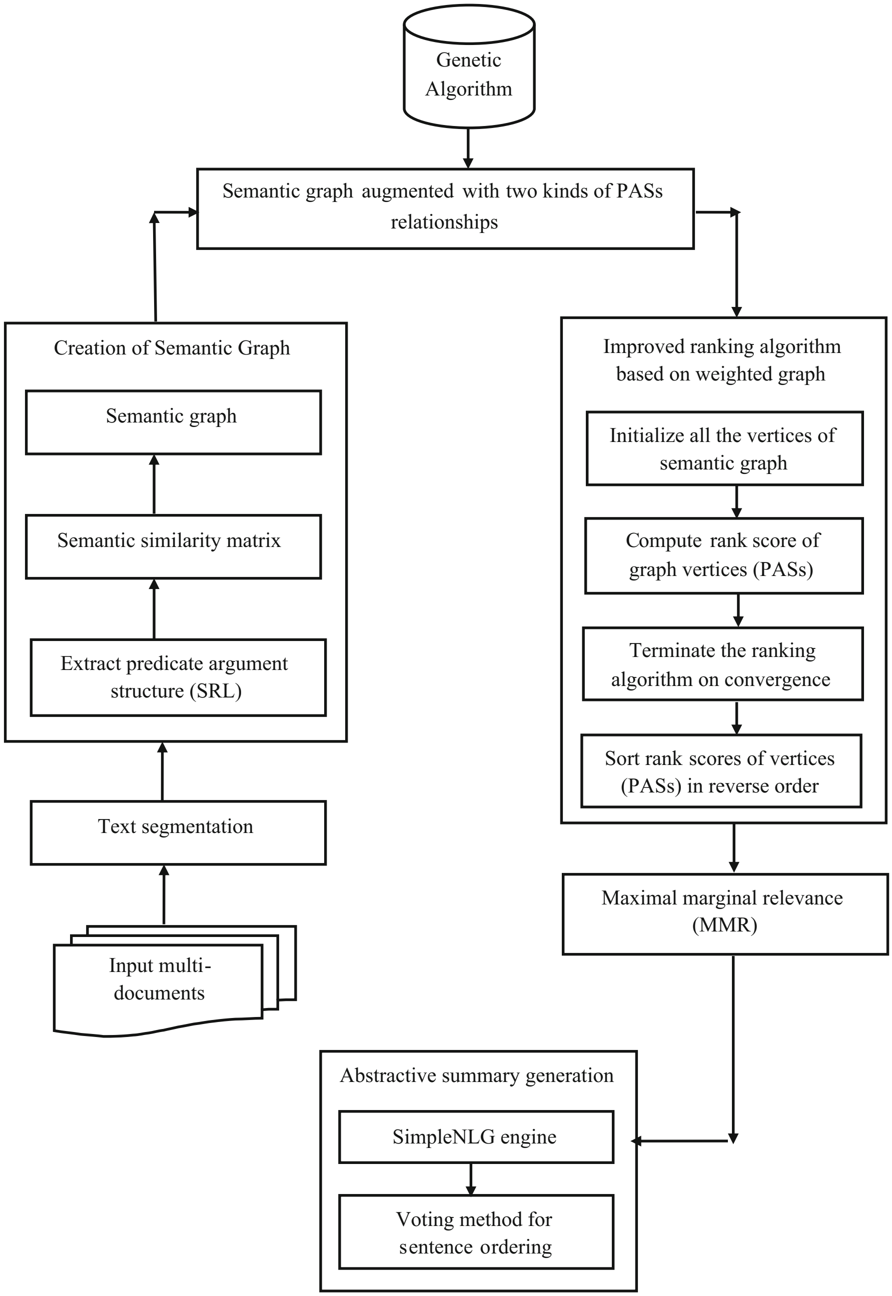
Genetic Algorithm
Semantic graph augmented with two kinds of PASs relationships
Creation of Semantic Graph
Semantic graph
Semantic similarity matrix
Extract predicate argument structure (SRL)
Text segmentation
Input multi-documents
Improved ranking algorithm based on weighted graph
Initialize all the vertices of semantic graph
Compute rank score of graph vertices (PASs)
Terminate the ranking algorithm on convergence
Sort rank scores of vertices (PASs) in reverse order
Maximal marginal relevance (MMR)
Abstractive summary generation
SimpleNLG engine
Voting method for sentence ordering

Fig. 1 Proposed semantic graph approach

sentences from the chosen graph vertices, which indicate the PASs (as discussed in Sect. 3.5).

3.1 Creation of Semantic Graph

This phase aims to build semantic graph from the document collection. The steps involved in this phase are as follows:

3.1.1 Semantic Role Labelling

The goal of this step is to parse each sentence in the document collection and extracts PAS from them. We begin by segmenting the document collection into sentences. Each sentence is given a key based on its document number and location number. As profound semantic text analysis is carried out in abstractive summarization, therefore, we employ SENNA semantic role parser [40,41] to determine PAS from the sentence collection by labeling the semantic phrases appropriately. The semantic phrases are also known as semantic arguments that can be organized in two sets: core arguments (Arg) and adjunctive arguments (ArgM) as depicted in Table 1.

For each predicate (Verb) V, this work assumes the following list of core arguments: $A0$(subject), $A1$(object), $A2$(indirect object), and adjunctive arguments: *ArgM-TMP*(time), *ArgM-LOC* (location). Moreover, our work simply presume the complete predicates linked with a sentence, in order to retain the salient terms and the real predicate of sentence.

Example 1 Consider the following two sentences represented by simple predicate argument structures.

S_1: "*Eventually, a huge cyclone hit the entrance of my house*".
S_2: "*Finally, a massive hurricane attack my home*".

After applying semantic role labeling to sentences S_1 and S_2, the corresponding simple predicate argument structures P_1 and P_2 are extracted are as follows:

P_1: *[AM-TMP: Eventually] [A0: a huge cyclone] [V: hit]* [A1: *the entrance of my house]*
P_2: *[AM-DIS: Finally] [A0: a massive hurricane] [V: attack] [A1: my home]*

When we achieve PASs, they are segmented into significant tokens/words, then stop words are taken out. Next, we make use of porter stemming algorithm [43] to

Table 1 Categories of semantic arguments

Core	Arguments	Adjunctive	Arguments
V	Verb	ArgM-DIR	Direction
A0	Subject	ArgM-MNR	Manner
A1	Object	ArgM-LOC	Location
A2	Indirect object	ArgM-TMP	Temporal marker
A3	Start point	ArgM-PRP	Purpose
A4	End point	ArgM-NEG	Negation
A5	Direction	ArgM-REC	Reciprocal
		AM-DIS	Discourse marker

transform the left over tokens in PASs to their core form. Afterwards, we utilize POS tagger [40] to assign Part of Speech (POS) tags to different tokens of the semantic arguments (linked with the predicates). The POS tags consists of verb (V), noun (*NN)*, adverb (*RB)*) and adjective (*JJ)* etc. Our work compare PASs based on verbs, nouns, time and location arguments. Hence, we get only words from the semantic arguments of PASs, which carry grammatical roles as verb, noun, time and location. The PASs in Example 1, after further processing is shown below:

P_1: *[AM-TMP: Eventually (SB)] [A0: cyclone (NN)] [VBD: hit] [A1: entrance (NN), house (NN)]*
P_2: *[A0: hurricane NN] [V: attack] [A1: home (NN)]*

3.1.2 Semantic Similarity Matrix

This step intends to calculate pair-wise semantic similarity scores PASs and then build a matrix from it. The verb, noun, time and location arguments of each PAS are compared with other PASs to compute pair wise similarities. Empirical findings revealed that Jiang similarity measure was found to have close agreement with humans as compare to other similarity measures. Consequently, this research make use of Jiang's measure [44] to find semantic similarity between PASs. This measure supposes that each concept that is present in the WordNet [45] own some information.

The measure exploits shared information of concepts while determining the similarity of any two concepts. Jiang's measure [44] finds the semantic distance of concepts, using the following equation:

$$Jiang_{dist}\,(C1,\ C2) = IC\,(C1) + IC\,(C2) - 2\,x\,IC\,(lso\,(C1, C2)) \qquad (1)$$

First, the Jiang's measure employs WordNet to find the least common subsumer (**lso**) of any two concepts, and that represents the shared parent of the given concepts in close proximity, then it determines $IC\,(C1)\,, IC\,(C2)\,, and\, IC\,(lso(C1,\ C2))\,.$ Information content (IC) of a concept can be approximated by calculating its likelihood of happening in a massive text corpus and is given as following equation:

$$IC\,(C) = -\log\ P(C) \qquad (2)$$

Where $P\,(C)$is the likelihood of existence of concept 'C'.

Formally, the semantic similarity of PASs p_i and p_j determined from any two sentences $S_i\, and\, S_j$, is denoted by $sim_{sem}\left(p_i, p_j\right)$ and is computed using Eq. (7); where $sim_{verb}\left(p_i,\ p_j\right)$ indicates predicates (verbs) similarity, computed using Eq. (4),$sim_{arg}\left(p_i, p_j\right)$ represents sum of semantic arguments similarities from corresponding PASs and is computed using Eq. (3). Jiang similarity measure is utilized to compute similarity of noun terms and verbs in the PASs. Similarity of time arguments is indicated by $sim_{tmp}\left(p_i,\ p_j\right)$ and is calculated using Eq. (5) and similarity of location arguments is denoted by$sim_{loc}\left(p_i, p_j\right)$ is determined using Eq. (6). We employ edit distance algorithm in Eqs. (5–6), to find the similarity of location and time arguments of PAS. The semantic similarity of any given two PASs is computed using Eqs. (3–7).

$$sim_{arg}\left(p_i,\ p_j\right) = sim\left(A0_i,\ A0_j\right) + sim\left(A1_i, A1_j\right) + sim\left(A2_i,\ A2_j\right) \quad (3)$$

$$sim_{verb}\left(p_i\,,\ p_j\right) = \left(sim(Verb_i,\ Verb_j)\right) \quad (4)$$

$$sim_{tmp}\left(p_i\,,\ p_j\right) = \left(sim(Tmp_i,\ Tmp_j)\right) \quad (5)$$

$$sim_{loc}\left(p_i\,,\ p_j\right) = \left(sim(Loc_i,\ Loc_j\,)\right) \quad (6)$$

Combining Eqs. (3), (4), (5), (6) to give equation (7):

$$sim_{sem}\left(p_i\,,\ p_j\right) = sim_{verb}\left(p_i\,,\ p_j\right) + [sim_{arg}\left(p_i\,,\ p_j\right) + sim_{tmp}\left(p_i\,,\ p_j\right) + sim_{loc}\left(p_i\,,\ p_j\right) \quad (7)$$

Once the similarity score for each PAS pair is gained, then semantic similarity matrix $M_{i,j}$ is created from the pair wise similarity scores. $M_{i,j}$ is illustrated as given:

$$M_{i,j} = \begin{cases} M_{sim(p_i,\,p_j)} & if\, i\neq j \\ 0 & else \end{cases} \quad (8)$$

Where $M_{sim}\left(p_i,\ p_j\right)$ indicates semantic similarity score of PASs p_i and p_j in the matrix $M_{i,j}$.

3.1.3 Semantic Graph

A semantic graph (an undirected weighted graph) is created from the similarity matrix (as discussed in Sect. 3.1.2) in a manner that if similarity weight $sim\left(p_i, p_j\right)$ between PASs (vertices) p_i and p_j is greater than zero, then link is set up between PASs; otherwise a link is not created. So the link or edge of graph represent similarity weight between PASs. This study defines a semantic similarity threshold i.e. $0 < \beta \leq 0.5$ based on empirical analysis [12]. Consequently, a link is created between PASs (vertices), if semantic similarity is within the range; otherwise no link is set up if similarity is outside the range.

We let $sim\left(p_i, p_j\right) = 0$ to avoid self-transitions. PASs with similarity greater than 0.5 are assumed to be semantically equivalent, and are not incorporated in the semantic graph so that redundant PASs are ignored in the summary generation phase. The similarity weight $sim\left(p_i, p_j\right)$ between PASs p_i and $p_j \quad (i \neq j)$ is determined based on Jiang semantic similarity measure. A sample semantic graph is shown in Fig. 2. The edges of the semantic graph are displayed with different solid bars, indicating similarity weights in different ranges. The similarity weight of two PASs v_i and v_j is calculated using Eq. (7), and is re-written as given below:

$$f\left(v_i, v_j\right) = sim_{sem}\left(v_i\,,\ v_j\right) \quad (9)$$

Where $sim_{sem}\left(v_i\,,\ v_j\right)$ refers to the semantic similarity weight between two PASs v_i and v_j.

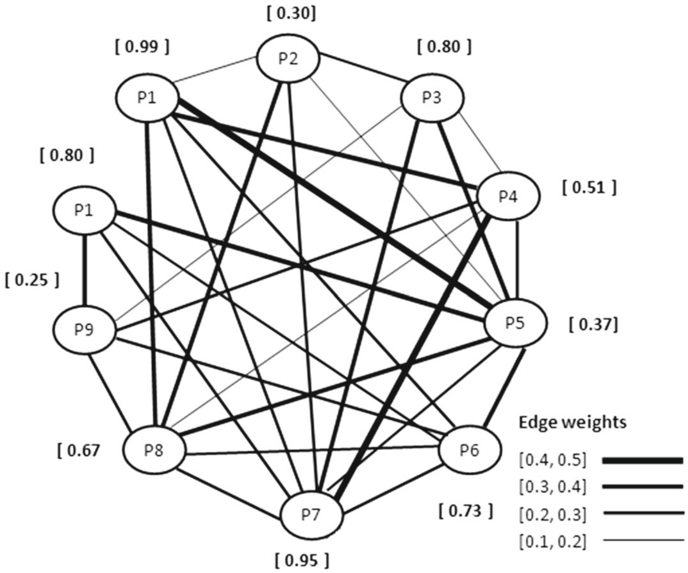

Fig. 2 PageRank score for different vertices of the semantic graph, and are given in square brackets close to the vertices

3.2 Semantic Graph Augmentation with PAS Relationships

Previous graph-based approaches treat sentences uniformly in the same and diverse documents, i.e. the sentences are rated without considering relationships of sentence to document and document set. However, this work assume PAS—the semantic structure of sentence; and besides PAS-to-PAS semantic similarity, we also examine the relative importance of two kinds of PAS relationships in the importance analysis of PASs.

In order to reveal the influence of both document and document set on PASs, the edge of semantic graph is augmented with PAS-to-document and PAS-to-document set relationships, whereas the graph edge represents semantic similarity weight between PASs.

We assume that the PASs which are more related with the document/document set, are more reasonable to be included in summary. Based on text summarization literature, we employ text features that are more relevant to represent predicate argument structure to document relationship, which includes: title feature [46], location [46] and PAS to document semantic similarity. Similarly, we also use text features that are more appropriate to signify the correlation/relationship of predicate argument structure to document set, and are given as follows: term weight [1] and frequent term [1] and PAS to document set semantic similarity.

Summary quality is vulnerable to text features i.e. different feature have not same importance in the process of producing summary. Therefore, assigning weights to features is essential for creating a good summary. This study make use of genetic algorithm (GA) to get optimal weights for different features. GA is selected since it is an optimization technique which is robust, and employed in diverse disciplines of applications and research, especially to deal with the problems of optimization [47]. Compared to other optimization techniques such as particle swarm optimization (PSO), GA has been well-known in academia and industry primarily because of its intuitiveness, ease of implementation, faster convergence and the capability to effectively solve a wide range of optimization problems [48]. GA is trained and tested on 59 data sets (fetched from DUC 2002), which are chosen by exploiting 10-fold cross validation. The GA procedure starts by setting the initial population to 50 chromosomes, which are assigned with the random real values in the range of 0 and 1. The fitness function indicates the average recall achieved from summaries of multiple documents. The fitness function computes the fitness values of chromosomes, which indicates how the good the chromosome is. When all the training multi-documents are summarized, then in this study, the best average recall will serve as a fitness function, which is computed from multi-documents summaries generated by ROUGE-1 [49], and is given in the following Equation.

$$F\left(X\right)=\frac{\sum_{S\in\{Reference\ Summaries\}}\sum_{gram_n\in S}Count_{match}\left(gram_n\right)}{\sum_{S\in\{Reference\ Summaries\}}\sum_{gram_n\in S}Count\left(gram_n\right)} \tag{10}$$

where n is the length of the *n-gram*, $gram_n$ and $count_{match}(gram_n)$ is the number of *n-grams* that simultaneously occur in a system summary and a set of human summaries. In this study, $n = 1$ is used.

The parents for next generation are chosen based on the fitness values of chromosomes existing in the current pool of population i.e. best chromosomes with high fitness values will be considered as parents. The reproduction operations of cross over and mutation operations are applied to chosen parent chromosomes to generate the new individual chromosomes. In this study, we implemented scattered method to execute the crossover operation, and Gaussian mutation method to accomplish the mutation operation.The GA process is usually converged when 100 maximum generations are reached and is therefore terminated. The individual chromosome with maximum fitness value represents optimal weights for features.

The scores of the aforementioned features altered with optimal weights are determined, and then merged to yield the relationship strength between PAS to document and PAS to document set. The relationship strength between PAS v_i and its corresponding document $doc\left(v_i\right)$ is given:

$$w\left(v_i,\ doc\left(v_i\right)\right)=\sum_{k=1}^{3}v_i\_f_k \tag{11}$$

Equation (11) is rewritten to indicate the features scores altered with optimal weights, which are achieved with GA and is shown below:

$$w\left(v_i,\ doc\left(v_i\right)\right)=\sum_{k=1}^{3}w_k\times\ v_i\_f_k \tag{12}$$

Where $w\left(v_i,\ doc\left(v_i\right)\right)$ indicates the relationship strength between PAS v_i and its corresponding document, $v_i\_f_k$ denotes feature k score for predicate v_i and w_k represents the feature k weight.

The relationship strength between PAS v_i and its corresponding document set $D_{set}\left(v_i\right)$ is given as follows:

$$w\left(v_i,\ D_{set}\left(v_i\right)\right) = \sum_{k=1}^{3} v_i\_f_k \tag{13}$$

Equation (13) is rewritten as follows, to indicate the features scores altered with optimal weights, which are achieved with GA.

$$w\left(v_i,\ D_{set}\left(v_i\right)\right) = \sum_{k=1}^{3} w_k \times\ v_i\_f_k \tag{14}$$

where $w\left(v_i,\ D_{set}\left(v_i\right)\right)$ denotes the relationship strength between PAS v_i and its corresponding document set $D_{set}\left(v_i\right)$.

Next, we additionally augment the edge of semantic graph with PAS-to-document and PAS-to-document set relationships, in order to reveal the influence of both document and document set on PASs. The graph edge represents semantic similarity weight between PASs. Thus, the new restricted similarity weight labeling the edge of the semantic graph is denoted by $f\left(v_i, v_j \mid doc\left(v_i\right),\ doc\left(v_j\right),\ D_{set}\left(v_i\right),\ D_{set}\left(v_j\right)\right)$. It is calculated by linearly integrating the similarity weight confined to the document containing the PAS v_i i.e. $f\left(v_i, v_j \mid doc\left(v_i\right)\right)$, the similarity weight restricted to another document including the PAS v_j i.e. $f\left(v_i, v_j \mid doc\left(v_j\right)\right)$, the similarity weight confined to the document set including the PAS v_i i.e.$f\left(v_i, v_j \mid D_{set}\left(v_i\right)\right)$ and similarity weight confined to the same document set including the PAS v_j i.e. $f\left(v_i, v_j \mid D_{set}\left(v_j\right)\right)$

The constrained similarity weight is precisely determined as given below:

$$\begin{aligned}
& f\left(v_i, v_j | doc\left(v_i\right),\ doc\left(v_j\right),\ D_{set}\left(v_i\right),\ D_{set}\left(v_j\right)\right) \\
& = \mu\, .f\left(v_i, v_j \mid doc\left(v_i\right)\right) + (1-\ \mu)\, .f\left(v_i, v_j \mid doc\left(v_j\right)\right) \\
& \quad + \mu\, .f\left(v_i, v_j \mid D_{set}\left(v_i\right)\right) \\
& \quad +\ (1-\ \mu)\, .f\left(v_i, v_j \mid D_{set}\left(v_j\right)\right) \\
& = \mu\, .f\left(v_i, v_j\right) .\, w\left(v_i,\ doc\left(v_i\right)\right) \\
& \quad + (1-\ \mu)\, .f\left(v_i, v_j\right)\ .\, w\left(v_j,\ doc\left(v_j\right)\right) \\
& \quad + \mu\, .f\left(v_i, v_j\right)\ .w\left(v_i,\ D_{set}\left(v_i\right)\right) +\ (1-\mu)\, .f\left(v_i, v_j\right)\ .\, w\left(v_j,\ D_{set}\left(v_j\right)\right) \\
& = f\left(v_i, v_j\right)\ .\, [\, \mu\, .w\ \left(v_i,\ doc\left(v_i\right)\right) \\
& \quad + (1-\ \mu)\ .\, w\left(v_j,\ doc\left(v_j\right)\right) + \mu\, .w\left(v_i,\ D_{set}\left(v_i\right)\right) \\
& \quad + (1-\mu)\ .\, w.\left(v_j,\ D_{set}\left(v_j\right)\right)] \\
& = f\left(v_i, v_j\right)\ .\, \mu\, .\, [\, w\ \left(v_i,\ doc\left(v_i\right)\right) \\
& \quad + w\left(v_i,\ D_{set}\left(v_i\right)\right)] +\ (1-\ \mu)\, .[\, w\left(v_j,\ doc\left(v_j\right)\right) \\
& \quad + w\left(v_j,\ D_{set}\left(v_j\right)\right)\,]
\end{aligned}$$

$$= sim_{sem}\left(v_i, v_j\right).\mu.\left[w\left(v_i, doc\left(v_i\right)\right) + w\left(v_i, D_{set}\left(v_i\right)\right)\right] \\ + (1-\mu).\left[w\left(v_j, doc\left(v_j\right)\right) + w\left(v_j, D_{set}\left(v_j\right)\right)\right] \quad (15)$$

where $\mu \in [0, 1]$is a combination weight that control the respective contribution from the first document containing v_i, the second document containing v_j and the document set. In this study, μ is set to 0.5 (optimal value), based on experimental observations.

We use semantic similarity matrix M to characterize the graph G, in which the value at position (i, j) corresponds to the similarity weight of edge existing in the semantic graph. Let $M = M_{i,j}$ is given as follows:

$$M_{i,j} = \begin{cases} f\left(v_i, v_j | doc\left(v_i\right), doc\left(v_j\right), D_{set}\left(v_i\right), D_{set}\left(v_j\right)\right) & iff\ v_i\ and\ v_j\ is\ linked\ and\ i \neq j \\ 0, & otherwise \end{cases} \quad (16)$$

We normalize $M_{i,j}$ to $\hat{M}_{i,j}$ as given, in order to make the summation of rows equal to 1.

$$\hat{M}_{i,j} = \begin{cases} M_{i,j} \Big/ \sum_{j=1}^{|v|} M_{i,j} & if\ \sum_{j=1}^{|v|} M_{i,j} \neq 0 \\ 0, & otherwise \end{cases} \quad (17)$$

3.3 Improved Weighted Graph-Based Ranking Algorithm

Conventionally, Google's PageRank [50] and HITS algorithm [51] are ranking algorithms that have been effectively employed in link analysis and social networks. PageRank algorithm attempted on undirected graph gave the best performance in DUC 2002 extractive summarization task for multi-documents.

The PageRank is a ranking algorithm that offers a way for determining the significance of a vertex in a graph by assuming the global information from the whole graph. Existing graph-based methods employ procedure similar to PageRank, and use content similarity rather than semantic similarity to determine relationships/associations between sentences. They treat sentences uniformly in the same and different documents. In other words, all sentences are rated without taking into consideration the sentence-to-document relationship and sentence-to-document set relationship. These approaches consider that the two types of sentence relationships have same importance, which is certainly unsatisfactory.

Based on our knowledge from literature, graph-based ranking algorithms have not been employed for MDAS. This work utilize an improved ranking algorithm based on weighted graph (*IWGRA*) as described in Algorithm 6.1. The ranking algorithm will consider edge weights in the importance analysis of vertices (PASs). The edge weight is computed from PAS-to-PAS semantic similarity, PAS-to-document and PAS-to-document set relationships. The importance score of vertex (PAS)v_i, denoted by $IWGRA\left(v_i\right)$, is concluded from all those vertices (PASs) that are linked to it and formulated in a recursive style as given in the following equation:

$$IWGRA\,(v_i) = (1 - d_p) + d_p . \sum_{v_j \in\, In(v_i)} \frac{IWGRA(v_j).w_{ji}}{\sum\limits_{v_z \,\in\, Out(v_j)} w_{zj}} \tag{18}$$

Where d_p indicates the damping factor and generally given a value of 0.85 [50] in the ranking model. $In\,(v_i)$ denotes the vertices that point to a given vertex v_i, $Out\,(v_j)$ represents set of links going out from *vertex* v_j, w_{ji} indicates the edge weight between vertices (PASs) v_i and v_j and w_{zj} denotes the weights of outgoing links from vertex v_j.

Algorithm 1 Improved weighted graph-based ranking algorithm

Step	Main process	Process detail
1	*Initialization*	Initialize the rank scores of all vertices of graph to 1. Set the damping factor to 0.85.
2	*Computation of rank score*	*2.1* Retrieve the vertices of the graph.
		2.2 For each vertex of the graph, determine the vertices that are linked to it.
		2.3 For each linked vertex, determine its importance by finding the number of outgoing links from that vertex, and then sum up the weights associated with the outgoing links.
		2.4 Compute the rank score of the given vertex considered in Step 2.2, using Eq. (18).
		2.5 Update the rank score of the vertex under consideration.
		2.6 Repeat Steps 2.2-2.5 until convergence is attained. Note convergence is attained when the difference between any two successive vertices scores falls below the given threshold (0.0001).
3	*Sort the rank scores of graph vertices*	Sort the rank scores obtained for the graph vertices in reverse order.

From the implementation perspective, the improved weighted graph-based ranking algorithm (IWGRA) begins by initializing the rank scores to all vertices (PASs) as 1, then the algorithm proceeds to determine the number of vertices that are linked to the current vertex under consideration. Once the number of linked vertices to the current given vertex are obtained, then the ranking algorithm calculates the importance of the linked vertices by finding the number of links going out from those vertices, and sum up the weights associated with the outgoing links. This means that the ranking algorithm computes the score of the given vertex, not exclusively based on the number of vertices that are linked to it, but also takes into account the salience of the linked vertices. After the importance of the linked vertices are determined, then the IWGRA employs Eq. (18), is run to find the new importance/rank scores of the vertices (PASs), as illustrated in Algorithm 1. The iteration algorithm recursively computes the importance scores of vertices till the convergence is attained. Fig. 2 depicts the rank/ importance scores of the graph vertices, achieved with the our improved ranking algorithm. The rank scores are enclosed in square brackets, and appear next to each vertex of the graph.

The ranking algorithm is converged, when rank scores difference for any vertices at two consecutive iterations fall down from the defined threshold (0.0001 in this study) [52]. Finally, the rank scores achieved for the vertices (PASs) of semantic graph are arranged in reverse order.

3.4 MMR to Control Redundancy

The ranking algorithm may assign same rank score to PASs representing the same concept and hence the final summary may contain redundancy. Furthermore, the other concepts of the documents which represents group of least similar PASs may not be included in the final summary, and thus significant information may lost. In this study, we employ a modified version of MMR [53] to reduce redundancy by re-ranking the PASs for inclusion in summary. A predicate argument structure is included in the summary generation list if it is not too similar to any existing PAS in the summary generation list. At first, the PAS with the maximum salience score is chosen and appended into the summary generation list. Then, the subsequent PAS having the higher salience score according to Eq. (18), is selected and added into summary generation list. This process chooses PASs by taking into account both importance and redundancy and keeps on repeating until the compression rate of summary is met.

$$MMR = argmax_{p_i \in R \backslash \mathrm{P}} \left[\alpha . RS\left(p_i\right)\right] - (1-\alpha) . \max_{p_j \in p} . sim\left(p_i, p_j\right) \tag{19}$$

In above Equation, R is the set of all predicate argument structures to be summarized, P is the set of PASs already chosen for inclusion in summary generation, $R \backslash P$ is the set of as yet unselected PASs in R, $RS\left(p_i\right)$ is the ranking (salience) score for PAS determined previously, $sim\left(p_i, p_j\right)$ refers to the semantic similarity between PASs, and α is a tuning parameter between PAS's importance and its relevance to previously chosen predicate argument structures. We set value of α= 0.6 [27] for the optimal performance. Once the graph vertices (predicate argument structures) are re-ranked using MMR, then the top scored vertices (PASs) are chosen based on compression rate of summary and are given to next phase that employs a SimpleNLG engine.

3.5 Summary Generation

In this phase, we make use of SimpleNLG [42] along with simple heuristic rules defined in it, to produce summary sentences from the top scored representative PASs chosen in previous phase. SimpleNLG engine utilizes simple English grammar rules to convert syntactical structures into sentences. The engine is robust i.e. it is not crashed, when partial syntactical structures are provided as input.

The first heuristic rule states that "*if the subjects in the predicate argument structures (PASs) refer to the same entity, then merge the predicate argument structures by removing the subject in all PASs except the first one, separating them by a comma (if there exist more than two PASs) and then combine them using connective 'and'* "

The second rule states that "*If PAS P_i subsumes a PAS P_j , then the subsumed PAS P_j is discarded in order to avoid redundancy*".

We discussed earlier that that this work assume specific arguments i.e. $A0$ for subject, $A1$ for object, $A2$ for indirect object as core arguments, and *ArgM-LOC* for location, *ArgM-TMP* for time as adjunctive arguments for predicate (Verb) V, while other arguments are throw away. Hence, the summary sentences created from PASs will be the compressed version of the original sentences in document collection. The heuristic rule defined in SimpleNLG merges the PASs that indicate the same subject. The example given below, describes the formation of abstractive summary sentence from the given source sentence using the heuristic rule.
For instance, the following source input sentences :

> S_1: "*Hurricane Gilbert claimed to be the most intense storm on record in terms of barometric pressure*".
> S_2: "*Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds* ".
> S_3: "*Hurricane Gilbert ripped roofs off homes and buildings*".

After applying SENNA SRL: the corresponding three predicate argument structures P_1, P_2 and P_3 are obtained as follows:

> P_1: *[A0: Hurricane Gilbert] [V: claimed] [A1: to be the most intense storm on record]*
> P_2: *[A0: Hurricane Gilbert] [V: slammed] [A1: into Kingston] [AM-TMP: on Monday]*
> P_3: *[A0: Hurricane Gilbert] [V: ripped] [A1: roofs off homes and buildings]*

From the previous step, P_1, P_2 and $P_3$are considered to be the top ranked PASs. After Applying the rule1 on the above example, we found that subject $A0$ is repeated and is ignored from all PASs apart from the first one. The SimpleNLG engine utilizes first heuristic rule on the three PASs given in aforementioned example, and creates the summary sentence, which is the condensed version of the original sentence in the document.
Summary Sentence: "Hurricane Gilbert claimed to be the most intense storm on record, slammed into Kingston on Monday with torrential rains and ripped roofs off homes and buildings".
After the generation of summary sentences, then we propose a voting scheme to re-arrange the summary sentences. The scheme operates in four steps:
Step 1: The voting scheme is based on the idea of voting or recommendation i.e. one sentence vote another sentence, if it addresses the same concept as described in another sentence, which is determined based on different similarity relations between them. In this work, a well know similarity measure, Jaccard similarity measure [54] is employed to calculate the similarity between a summary sentence and source sentences in document set. A scoring threshold (β) is assigned a value of 0.5 for sentence similarity, according to the literature [55]. Jaccard measure for sentences S_1 and S_2 is stated as the following equation.

$$J\left(s_1, s_2\right) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|} \tag{20}$$

Step 2: Once the similar source sentences (contained in the document set) to the corresponding summary sentence are identified, then the positions of the source sentences are averaged to give score to the corresponding summary sentence. We exploit position feature to determine the normalize position [0,1] of a source sentence in the document set using the following equation.

$$\begin{aligned} &Position\ of\ \ Source\ Sentence \\ &\quad = \frac{Document\ length\ -\ Sentence\ position\ in\ doc + 1}{Document\ length} \end{aligned} \tag{21}$$

Step 3: Repeat step 1 and step 2 until scores are assigned to all summary sentences.
Step 4: Arrange the sentences in the summary according to ascending order of scores.

4 Evaluation Results

The proposed improved semantic graph approach for MDS is assessed using DUC 2002 data sets (DUC, 2002). In the text summarization community, DUC 2002 is regarded as a benchmark dataset that includes documents along with the human made extractive and abstractive summaries. We chose this dataset as our work take into consideration multi-document extractive and abstractive summarization tasks. Other DUC datasets are not considered, as they do not have human made abstracts.

This work make use of both ROUGE [49] and Pyramid [56] evaluation metrics for the evaluation of our proposed approach. Pyramid evaluation metric is utilized is to compare our proposed semantic approach (Sem-Graph-Both-Rel) with the latest abstractive approach for multi-document summarization (AS) [16], average of automatic systems, best system, and average of human made summaries, in the context of MDAS shared task in DUC 2002. On other hand, ROUGE-1 and ROUGE-2 evaluation measures are employed to compare our approach (Sem-Graph-Both-Rel) with graph based MDES approach (Event graph), best system, and average of human made summaries, in the context of DUC 2002 MDES shared task.

Mean Coverage Score (MCS) or Pyramid score [56] for candidate summary is computed as:

$$MCS = \frac{Sum\,of\ \ SCUs\ Weight\ in\ candidate\ summary}{Average\ SCU\ in\ the\ reference\ Summary} \tag{22}$$

Where SCUs indicate content units in summary and SCUs weights refer to the number of human made summaries the SCU exist in.

The precision (P) for candidate summary [56] is determined as:

$$P\ = \frac{Number\,of\ \ Reference\ SCUs\ in\ candidate\ Summary}{Average\ SCU\ in\ the\ candidate\ Summary} \tag{23}$$

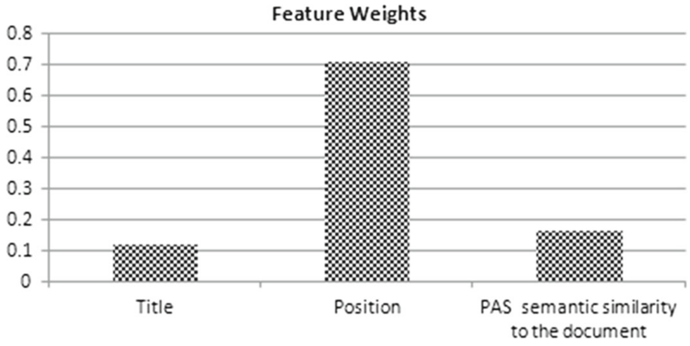

Fig. 3 Optimal weights for features achieved with GA for PAS-to-document relationship

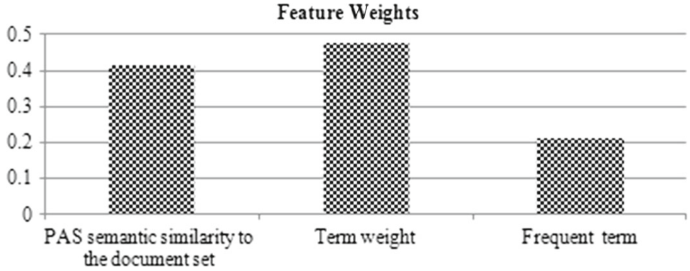

Fig. 4 Optimal weights for features attained with GA for PAS-to-document set relationship

The F-measure (F) for candidate summary is computed from Eqs. (22) and (23) as:

$$F = \frac{2 \times MCS \times P}{MCS + P} \tag{24}$$

ROUGE − N [49] is stated as an n-gram recall between a candidate summary and a set of human generated summaries

As discussed earlier, the augmented two kinds of PAS relationships in the semantic graph i.e. PAS-to-document and PAS-to-document set, are represented by various features. So, the results of optimal weights for features achieved with GA for both kinds of relationships are depicted in Figs. 3 and 4. The optimal weights achieved are 0.131493, 0.7238985, 0.1737921,which represent the weights for different features such as title, position and PAS semantic similarity to the document respectively.

Similarly, in the case of PAS-to-document set relationship, the optimal weights attained are 0.413137516, 0.475472113, 0.213086421 representing the weights for different features such as PAS semantic similarity to the document set, term weight and frequent term respectively.

To evaluate the effectiveness of our technique (Sem-Graph-Both-Rel) in the context of DUC 2002 MDAS task, five comparison models (Avg, Best, AS, Only PAS-Docset-Rel, Only PAS-Doc-Rel) were set up, besides the human summaries. Table 2 illustrates the evaluation results for abstractive summaries obtained by different systems over

Table 2 MDAS results for different systems based on pyramid evaluation measures

System	Mean coverage score	AVG-precision	AVG-F-Measure
Models	0.6910	0.8528	0.7634
Sem-graph-both-rel	0.5441	0.7563	0.6396
Only PAS-docset-rel	0.5247	0.7267	0.6094
Only PAS-doc-rel	0.5100	0.7110	0.5940
AS (Genest and Lapalme 2011)	0.4378	0.59	0.50
Best (System code:19) [17]	0.2783	0.7452	0.4053
Avg	0.1775	0.6700	0.2806

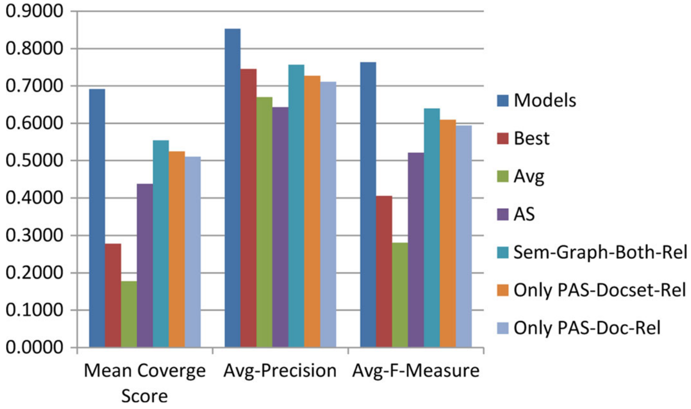

Fig. 5 MDAS results for different systems based on Pyramid evaluation measures

Pyramid evaluation metrics on DUC 2002 dataset. Figure 5 visualizes the results of abstractive summaries attained with proposed approach and other models.

There are three variants of our semantic graph based abstractive summarization approach that will be evaluated: the complete summarization approach (SemGraph-Both-Rel), which takes into account both **PAS-document** relationship and **PAS-document set** relationship, besides PAS to PAS semantic similarity. The other two variants of our summarization approach: "Only PAS-Docset-Rel" and "Only PAS-Doc-Rel" employ PAS to PAS semantic similarity but are differentiated by the type of relationship they hold i.e. the former employs PAS-document set relationship across different but topically related documents. While the latter employs PAS-document relationship within the specified document. Best (System 19) is the top ranked system for creating abstracts, Avg denotes the average of abstractive summarization systems contributing in DUC 2002, and the Models denote the average of human made abstractive summaries.

On the other hand, ROUGE-1 and ROUGE-2 evaluation measures are utilized to compare our approach (Sem-Graph-Both-Rel) with five comparison models evaluated in the context of MDES task defined for DUC 2002. The above Table 3 presents recall of extractive summaries on DUC 2002 data sets, obtained with proposed approach and

Table 3 MDES results for different systems based on ROUGE measures

System	Extractive/abstractive	ROUGE-1	ROUGE-2
DUC-2002 Models	Extractive	0.418	0.102
Sem-graph-both-rel	Abstractive	0.417	0.108
Event graph [29]	Extractive	0.415	0.116
Only PAS-docset-rel	Abstractive	0.410	0.101
Only PAS-doc-rel	Abstractive	0.40	0.099
DUC-2002 Best (system 21) [57]	Extractive	0.395	0.103

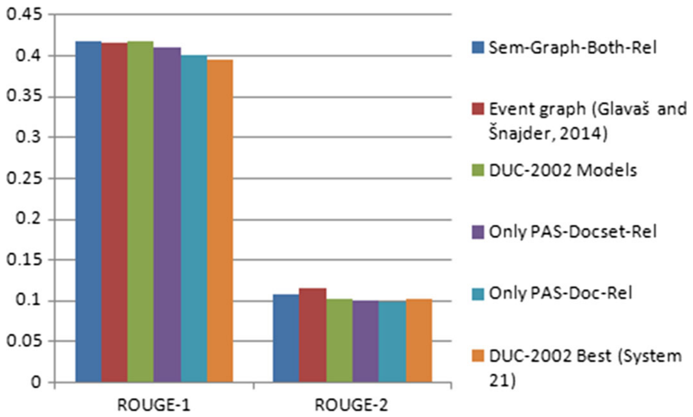

Fig. 6 MDES results for different systems based on ROUGE measures

other summarization models. The recall of summaries was obtained with ROUGE-1 and ROUGE-2 measures. Figure 6 depicts the results of extractive summaries attained with the proposed approach and other extractive summarization models.

4.1 Discussion

In this section, the achieved experimental results are discussed in detail. At first, the proposed method is compared with other summarization models based on three pyramid evaluation metrics. The results from Table 2 demonstrate that the proposed approach (Sem-Graph-Both-Rel) outperforms the comparison models over pyramid evaluation metrics; and appeared second to Models, which represents the mean of human made summaries. The summarization results achieved for the proposed approach and other comparison models are also validated by performing statistical test (Paired-Samples T-test), and our approach obtained a lower significance value of T < 0.05. The experimental findings recommend that semantic graph is a suitable representation for abstractive summary generation, and the improved ranking algorithm (based on weighted graph) in the proposed approach significantly improves the summarization results. Thus, the experimental results support the claim that MDAS using semantic graph enhance the summarization results. Furthermore, Table 3 demonstrates the ROUGE results for the proposed approach and other comparison systems. Our approach (Sem-Graph-Both-Rel) performs superior than the best system and graph-based MDES approach (Event graph) on ROUGE-1 metric. However, the proposed approach showed inferior performance as compared to other systems on ROUGE-2 metric. This is due to the fact that our approach generates summary that contains condensed version of original sentences in the document; while on other hand, extractive summarization systems generate summary that contains original source sentences. ROUGE measures look for exact matches of text snippets while comparing system summary against human produced summary (extracts). Thus, the abstractive summary produced by our approach will contain lesser matching text snippets as compared to event graph-based extractive summarization system.

Furthermore, in order to investigate the impact of two kinds of relationships (PAS-to-document and PAS-to-document set) on the performance of the proposed semantic approach, we dropped one relationship and consider the other, and vice versa. The "Only PAS-Doc-Rel" represents the semantic graph-based approach based only on PAS-to-document relationship, while the "Only PAS-Docset-Rel" represents the semantic graph-based approach based only on PAS-to-document set relationship. Sem-Graph-Both-Rel represents our semantic approach that take into consideration both kinds of relationships.

From the perspective of both pyramid and ROUGE evaluation results presented in Tables 2 and 3, the proposed semantic graph-based summarization approach (Sem-Graph-Both-Rel) performs better than its two variants: the Only PAS-Docset-Rel and "Only PAS-Doc-Rel". These results reveal that both kinds of relationships are important as they play significant role in the ranking process of important graph vertices (PASs); and thus contribute in the performance of the proposed semantic graph for MDS. Besides the two types of relationships in the proposed semantic approach (Sem-Graph-Both-Rel), PAS to PAS semantic similarity also contribute in the performance of the semantic approach. It facilitates the graph ranking algorithm in selection of high ranked representative graph vertex (PAS) by capturing its votes from other graph vertices (PASs) that are semantically related to it. Moreover, we can see that the method "Only PAS-Docset-Rel" based only on PAS-to-document set relationship outperforms the method "Only PAS-Doc-Rel" relying only on PAS-to-document relationship, which further reveals the great significance of PAS-to-document set relationship for multi-document summarization.

5 Conclusion and Future Work

Even though abstractive summarization in real sense is hard to achieve, our semantic graph approach demonstrates the viability of this new track for summarization community. Existing graph based approaches consider sentence as vector of words and cannot detect semantically equivalent redundant sentences, as they mostly rely on content similarity measure. We integrate semantic similarity in the graph-based approach to determine semantic relationship between PASs, which will assist in detecting redundancy. Moreover, our approach improves graph ranking algorithm by incorporating

PASs semantic similarity, and the two kinds of PAS relationships. Experimental findings confirm that summary results are improved with the ranking algorithm. The approach is not dependent on humans and can be adapted to other domain. In future, we will explore how concept hierarchy or taxonomy (an alternative to semantic representation of document text), learnt automatically from text corpus can be utilized to generate abstractive summary. We suggest that formal context analysis (FCA) method can play significant role in building concept hierarchy from text corpus automatically. Once the concept hierarchy is constructed, concept relevance measure may be exploited to determine the relevance of concepts; and the most relevant concepts with the underlying relationships can be fed to a language generator to construct summary sentences.

Acknowledgements This research is supported by Higher Education Commission (HEC), Pakistan and Department of Computer Science, Islamia College, Peshawar, Pakistan. This research is also supported by Next-Generation Information Computing Development Program through the National Research Foundation (NRF) funded by the Korean Government (MSIT) (2017M3C4A7066010). This work is also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (NRF2016R1A2A1A05005459).

References

1. Fattah, M.A., Ren, F.: GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Comput. Speech Lang. **23**(1), 126–144 (2009)
2. Barzilay, R., McKeown, K.R.: Sentence fusion for multidocument news summarization. Comput. Linguist. **31**(3), 297–328 (2005)
3. Das, D., Martins, A.F.: A survey on automatic text summarization. Lit. Surv. Lang. Stat. II course at CMU **4**, 192–195 (2007)
4. Ye, S., Chua, T.-S., Kan, M.-Y., Qiu, L.: Document concept lattice for text understanding and summarization. Inf. Process. Manag. **43**(6), 1643–1662 (2007)
5. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
6. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, USA, 9–13 July 1995, pp. 68-73. ACM (1995)
7. Knight, K., Marcu, D.: Statistics-based summarization-step one: Sentence compression. In: Proceedings of the National Conference on Artificial Intelligence 2000, pp. 703–710. AAAI Press, Menlo Park (1999)
8. Larsen, B.: A trainable summarizer with knowledge acquired from robust NLP techniques. Adv. Autom. Text Summ. **71** (1999)
9. Fattah, M.A.: A hybrid machine learning model for multi-document summarization. Appl. Intell. **40**(4), 592–600 (2014)
10. Erkan, G., Radev, D.R.: LexPageRank: prestige in multi-document text summarization. In: EMNLP 2004, pp. 365–371 (2004)
11. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. (JAIR) **22**(1), 457–479 (2004)
12. Mihalcea, R., Tarau, P.: A language independent algorithm for single and multiple document summarization (2005)
13. Wan, X., Yang, J.: Improved affinity graph based multi-document summarization. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, New York City, USA, June 2006, pp. 181–184. ACL (2006)
14. Barzilay, R., McKeown, K.R., Elhadad, M.: Information fusion in the context of multi-document summarization. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, 20–26 June 1999, pp. 550–557. ACL (1999)

15. Tanaka, H., Kinoshita, A., Kobayakawa, T., Kumano, T., Kato, N.: Syntax-driven sentence revision for broadcast news summarization. In: Proceedings of the 2009 Workshop on Language Generation and Summarisation, Suntec, Singapore, 6 August 2009, pp. 39–47. ACL (2009)
16. Genest, P.-E., Lapalme, G.: Framework for abstractive summarization using text-to-text generation. In: Proceedings of the workshop on monolingual text-to-text generation, Oregon, USA, 24 June 2011, pp. 64–73. ACL (2011)
17. Harabagiu, S.M., Lacatusu, F.: Generating single and multi-document summaries with gistexter. In: Document Understanding Conferences, Pennsylvania, USA, 11–12 July 2002, pp. 40–45. NIST (2002)
18. Genest, P.-E., Lapalme, G.: Fully abstractive approach to guided summarization. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Jeju Island, Korea, 8–14 July 2012, pp. 354–358. ACL (2012)
19. Lee, C.-S., Jian, Z.-W., Huang, L.-K.: A fuzzy ontology and its application to news summarization. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **35**(5), 859–880 (2005)
20. Greenbacker, C.F.: Towards a framework for abstractive summarization of multimodal documents. ACL HLT **2011**, 75 (2011)
21. Moawad, I.F., Aref, M.: Semantic graph reduction approach for abstractive text summarization. In: 7th International Conference on Computer Engineering andSystems (ICCES), 2012, pp. 132–138. IEEE (2012)
22. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. (1999)
23. Mani, I., Bloedorn, E.: Summarizing similarities and differences among related documents. Inf. Retr. **1**(1–2), 35–67 (1999)
24. Zhang, J., Sun, L., Zhou, Q.: A cue-based hub-authority approach for multi-document text summarization. In: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, IEEE NLP-KE'05, 2005, pp. 642–645. IEEE (2005)
25. Wei, F., Li, W., Lu, Q., He, Y.: A document-sensitive graph model for multi-document summarization. Knowl. Inf. Syst. **22**(2), 245–259 (2010)
26. Ge, S.S., Zhang, Z., He, H.: Weighted graph model based sentence clustering and ranking for document summarization. In: 4th International Conference on Interaction Sciences (ICIS), 2011, pp. 90–95. IEEE (2011)
27. Nguyen-Hoang, T.-A., Nguyen, K., Tran, Q.-V.: TSGVi: a graph-based summarization system for Vietnamese documents. J. Ambient Intell. Humaniz. Comput. **3**(4), 305–313 (2012)
28. Cheung, J.C.K., Penn, G.: Towards Robust Abstractive Multi-Document Summarization: A Caseframe Analysis of Centrality and Domain. In: ACL (1), pp. 1233–1242 (2013)
29. Glavaš, G., Šnajder, J.: Event graphs for information retrieval and multi-document summarization. Expert Syst. Appl. **41**(15), 6904–6916 (2014)
30. Liu, F., Flanigan, J., Thomson, S., Sadeh, N., Smith, N.A.: Toward abstractive summarization using semantic representations. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, 1–5 June 2015, pp. 1077–1086. ACL (2015)
31. Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., Passonneau, R.J.: Abstractive multi-document summarization via phrase selection and merging. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015, pp. 1587–1597. ACL (2015)
32. Boudin, F., Mougard, H., Favre, B.: Concept-based summarization using integer linear programming: from concept pruning to multiple optimal solutions. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1914–1918. ACL (2015)
33. Belkebir, R., Guessoum, A.: Concept generalization and fusion for abstractive sentence generation. Expert Syst. Appl. **53**, 43–56 (2016)
34. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**(1), 1–66 (2017)
35. Cuomo, S., De Michele, P., Piccialli, F., Galletti, A., Jung, J.E.: IoT-based collaborative reputation system for associating visitors and artworks in a cultural scenario. Expert Syst. Appl. **79**, 101–111 (2017)
36. Farina, R., Cuomo, S., De Michele, P., Piccialli, F.: A smart GPU implementation of an elliptic kernel for an ocean global circulation model. Appl. Math. Sci. **7**(61–64), 3007–3021 (2013)

37. Piccialli, F., Cuomo, S., De Michele, P.: A regularized mri image reconstruction based on hessian penalty term on CPU/GPU systems. Proc. Comput. Sci. **18**, 2643–2646 (2013)
38. Chianese, A., Marulli, F., Moscato, V., Piccialli, F.: A "smart" multimedia guide for indoor contextual navigation in cultural heritage applications. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2013, pp. 1–6. IEEE (2013)
39. Chianese, A., Piccialli, F.: SmaCH: a framework for smart cultural heritage spaces. In: 10th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS), 2014, pp. 477–484. IEEE (2014)
40. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
41. Barnickel, T., Weston, J., Collobert, R., Mewes, H.-W., Stümpflen, V.: Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. PLoS ONE **4**(7), e6393 (2009)
42. Gatt, A., Reiter, E.: SimpleNLG: a realisation engine for practical applications. In: Proceedings of the 12th European Workshop on Natural Language Generation, Athens, Greece, 30–31 March 2009, pp. 90–93. ACL (2009)
43. Porter, M.F.: Snowball: a language for stemming algorithms (2001)
44. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. Preprint arXiv:cmp-lg/9709008 (1997)
45. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
46. Suanmali, L., Salim, N., Binwahlan, M.S.: Fuzzy logic based method for improving text summarization. Int. J. Comput. Sci. Inf. Secur. **2**(1), 65–70 (2009)
47. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**(6), 17–26 (1994)
48. Panda, S., Padhy, N.P.: Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. Appl. Soft Comput. **8**(4), 1418–1427 (2008)
49. Lin, C.-Y.: Rouge: a package for automatic evaluation of summaries. In: Proceedings of the ACL-04 workshop ontext summarization branches out, Barcelona, Spain, 25–26 July 2004, pp. 74–81. ACL (2004)
50. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. ISDN Syst. **30**(1), 107–117 (1998)
51. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
52. Mihalcea, R., Tarau, P.: A language independent algorithm for single and multiple document summarization. http://digital.library.unt.edu/ark:/67531/metadc30965/ (2005)
53. 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, Melbourne, Australia, 24–28 August 1998, pp. 335–336. ACM (1998)
54. Jaccard, Paul: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles **37**, 547–579 (1901)
55. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI 2006, pp. 775–780 (2006)
56. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: the pyramid method. In: 2004. NAACL-HLT (2004)
57. Over, P., Liggett, W.: Introduction to DUC: an intrinsic evaluation of generic news text summarization systems. http://www-nlpir.nist.gov/projects/duc/pubs/2002slides/overview.02.pdf (2002)