Jurnal
Teknologi

Full Paper

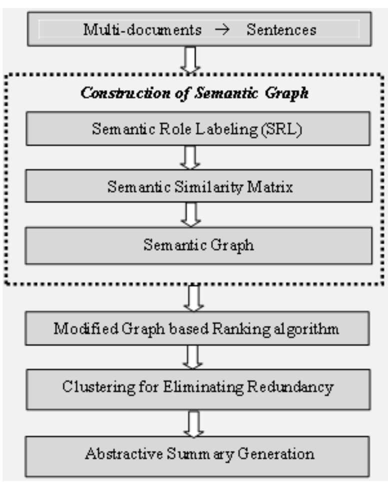
A Clustered Semantic Graph Approach for Multi-document Abstractive Summarization

Article history
Received
15 May 2015
Received in revised form
1 July 2015
Accepted
11 August 2015

Atif Khan*, Naomie Salim[a], Waleed Reafee[a], Anupong Sukprasert[a], Yogan Jaya Kumar[b]

[a]Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[b]Faculty of Information and Communication Technology, University Teknikal Malaysia Melaka, 76100 Melaka, Malaysia

*Corresponding author
atifkhan@icp.edu.pk



Abstract

Multi-document abstractive summarization aims is to create a compact version of the source text and preserves the important information. The existing graph based methods rely on Bag of Words approach, which treats sentence as bag of words and relies on content similarity measure. The obvious limitation of Bag of Words approach is that it ignores semantic relationships among words and thus the summary produced from the source text would not be adequate. This paper proposes a clustered semantic graph based approach for multi-document abstractive summarization. The approach operates by employing semantic role labeling (SRL) to extract the semantic structure (predicate argument structures) from the document text. The predicate argument structures (PASs) are compared pair wise based on Lin semantic similarity measure to build semantic similarity matrix, which is thus represented as semantic graph whereas the vertices of graph represent the PASs and the edges correspond to the semantic similarity weight between the vertices. Content selection for summary is made by ranking the important graph vertices (PASs) based on modified graph based ranking algorithm. Agglomerative hierarchical clustering is performed to eliminate redundancy in such a way that representative PAS with the highest salience score from each cluster is chosen, and fed to language generation to generate summary sentences. Experiment of this study is performed using DUC-2002, a standard corpus for text summarization. Experimental results reveal that the proposed approach outperforms other summarization systems.

Keywords: Multi document abstractive summarization, semantic role labeling (SRL), graph based ranking algorithm, semantic graph, semantic similarity measure

1.0 INTRODUCTION

Information on World Wide Web is growing at exponential speed and has increased the demand for automatic text summarization systems. In the current age of information overload, text summarization is a significant and timely tool for user to swiftly comprehend the massive volume of information. The goal of automatic text summarization is to extract the most salient content from the given source text and produce a compressed summary that can satisfy user's needs [1, 2]. Text summarization approaches can be broadly separated into two groups: extractive summarization and abstractive summarization. The goal of Extractive summarization is to extract the most important representative sentences from the source documents, and group them to produce a summary. However, abstractive summarization requires natural language processing techniques such as semantic representation, natural language generation, and compression techniques. Abstractive summarization aims to interpret and examines the source text and create a concise summary that usually contain compressed sentences or may contain some novel sentences not present in the original source text [3, 4].

Multi-document summarization has attracted more attention in recent years. Different approaches for multi-document summarization have been developed. Most of the studies have focused on multi-document extractive summarization, using techniques of sentence extraction [5], statistical analysis [6], discourse structures and various machine learning techniques [7, 8]. Different graph-based methods [9-12] have also been investigated for multi-document extractive summarization. These methods employ PageRank algorithm[13] or its variants for computing the relative importance of sentences.

However, abstractive summarization is a challenging area for researchers. To date, a few research efforts have been done in this direction. Two mainstream approaches are applied to multi-document abstractive summarization: Linguistic and semantic based approaches. Linguistic based approaches proposed for abstractive summarization employ tree based method [2, 14], lead and body phrase method [15] and information item based method [16]. On other hand, semantic based approaches proposed for abstractive summarization mainly employ template based methods [17, 18] and ontology based methods [19-21].

A particular challenge for multi-document summarization is that there is an inevitable overlap in the information content stored in different documents. Thus, effective summarization methods that merge similar information content across different documents are required [2]. In this connection, numerous methods have been devised but suffered from some limitations. In particular, the aforementioned graph based methods attempted for multi-document extractive summarization did not consider the semantic structure of sentence, treat sentence as bag of words and use content similarity measure for determining sentence similarity, which may fail to detect redundant sentences that are semantically equivalent. Thus, the final summary will contain redundant information. On other hand, the graph based abstractive approach presented by [21] constructs semantic graph from manually built ontology. This approach heavily relies on human expert and is limited to single document.

To our knowledge, semantic graph based method has not been explored for multi-document abstractive summarization. Thus, this study aims to propose a semantic graph based approach for multi-document abstractive summarization that attempts to overcome the limitations in the existing graph based approaches. The approach will automatically merge similar information across the documents, and employs language generation to generate abstractive summary. The approach integrates semantic role labeling (SRL) technique with graph to build semantic graph representation from the document text. The nodes (vertices) of graph represent predicate argument structures (PASs) and a link is established between PASs if their similarity weight exceeds zero, otherwise no link is established. The PASs are extracted from the source document text by employing semantic role labeling (SRL). The similarity weights among PASs are computed based on Lin semantic similarity measure [22]. The salience (importance) score of graph nodes (PASs) is determined based on modified graph based ranking algorithm (i.e. we incorporated PAS-PAS semantic similarity into the graph-based ranking algorithm), and finally the graph nodes (PASs) are sorted in reverse order based on salience scores. We apply agglomerative hierarchical clustering to eliminate redundant PAS in a manner that we select the most representative PAS (the one with the highest salience score) is chosen from each cluster. The representative PASs are then sorted based on their salience scores and are given to summary generation phase to produce summary sentences. Our contributions are summarized as follows:

- Introduce a semantic graph approach for multi-document abstractive summarization.
- To propose a domain independent approach for multi-document abstractive summarization.
- Modify graph based ranking algorithm to take into account the semantic similarity measure instead of content similarity measure.
- To evaluate the proposed semantic graph based approach with Pyramid and ROUGE evaluation measures on DUC 2002 multi-document summarization shared tasks.

The rest of this paper is organized as follows: Section 2 demonstrates the related work to this research study. Section 3 outlines the proposed approach. Section 4 presents the evaluation results and discussion. Finally we end with conclusion in Section 5.

2.0 RELATED WORK

This section is organized in the following manner: At first, we present multi-document abstractive summarization approaches, then we discuss graph based approaches introduced for multi-document extractive summarization and single document abstractive summarization. Finally, we briefly present our proposed semantic graph based approach for multi-document abstractive summarization.

Limited research studies have dealt with multi-document abstractive summarization and introduced linguistic (syntactic) and semantic based approaches for such type of summarization. Abstractive summarization based on linguistic approaches [2, 14-16] depends on syntactic representation of the source document, and hence these approaches lack of semantic representation of source text. A few semantic approaches have also been examined for multi-document abstractive summarization and are discussed as follows.

A multi-document summarization system, GISTEXTER, introduced by [17] employed template based representation of documents to produce abstractive summary from multiple newswire/newspaper documents. The obvious drawback of this approach was that linguistic patterns and extraction rules for template slots were manually created by humans, which is time consuming. Moreover, this method could not handle or capture similar information across multiple documents. A fuzzy ontology based approach [19] was proposed for Chinese news summarization to model uncertain information and hence can better describe the domain knowledge. This approach suffers from several limitations. First, domain ontology and Chinese dictionary need to be defined by a domain expert, which require more effort and is time consuming. Secondly, this approach is limited to Chinese news, and might not be applicable to English news. A framework proposed by [20] generates abstractive summary from a semantic model of a multimodal document. The semantic model is built from the concepts organized in ontology. However, the framework heavily relies on domain expert to build domain ontology, and is not applicable to other domains. The methodology presented by [18] employed abstraction schemes to generate short abstractive summaries from clusters of news articles on same event. The shortcoming of the methodology was that information extraction (IE) rules and generation patterns for abstraction schemes were written by hand, which required human expert knowledge. A series of analysis studies is performed by [23] to compare human-written model summaries with system summaries at semantic level of caseframes. However, these studies did not propose any summarization model.

In recent years, different graph based methods have been employed for multi-document extractive summarization. These methods use PageRank algorithm [13] or its variants to rank sentences or passages. [24] uses a graph connectivity model and manipulate under the assumption that nodes which are linked to many other nodes are probably to carry relevant information. The approach in [9] employed the concept of eigenvector centrality to compute the significance of sentence. It constructs a sentence connectivity matrix and uses algorithm similar to PageRank to determine the important sentences. An algorithm based on PageRank is proposed by [11] to determine salient sentences for document summarization. [25] introduced a graph based approach under a hub-authority framework, 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 multi-document summarization presented by [12] employs similar algorithm to PageRank to compute information richness scores of sentences in the affinity graph. The approach differentiated intra-document and inter-document links between sentences. However, all these graph methods discussed so far did not consider semantic links between sentences. [26] presented a document-sensitive graph model for multi-document generic summarization and highlights the impact of global document set information at sentence level. However, the model lacks semantic relationships between sentences. A weighted graph model for generic multi-document summarization introduced by [27] combines sentence ranking and sentence clustering methods. However, this approach did not take into account semantic relationship between sentences. [28] introduced a graph based method for multi-document summarization of Vietnamese documents and employed traditional PageRank algorithm to rank the important sentences. However, semantic similarity methods are not applicable to Vietnamese documents due to the lack of lexical resources such as English WordNet. [29] demonstrated an event graph based approach for multi-document extractive summarization. However, the approach requires the construction of hand crafted rules for argument extraction, which is a time consuming process and may limit its application to a specific domain. The only graph based approach introduced for abstractive summarization [21] constructs semantic graph from manually built ontology. This approach heavily relies on human expert and is limited to single document

The aforementioned graph based models are proposed for multi-document extractive summarization and treat sentence as a bag of words without understanding the meaning of sentence. These models employ traditional cosine similarity for determining sentence similarities, which may fail in detecting redundant sentences that are semantically equivalent, and therefore the final summary would be inadequate by having redundant sentences.

To our knowledge, clustered semantic graph approach has not been utilized for multi-document abstractive summarization (MDAS). Therefore, this study aims to introduce a clustered semantic graph approach for MDAS. Our proposed approach is different from existing graph based approaches in the following manner: First, we build semantic graph by integrating SRL with graph that can be applied to any domain, and does not require any intervention of human expert. Secondly, our approach extract predicate argument structure (PAS) from each sentence in the document text in order to capture the semantics of sentence (e.g. who did what to whom, when and how). Thirdly, we determine semantic relationships between predicate argument structures based on Lin semantic similarity measure [22]. Finally, the modified ranking algorithm (which takes into account the semantic similarity between PASs) is employed to determine the salience score of PASs.

3.0 OVERVIEW OF APPROACH

The architecture of our proposed approach is depicted in Figure 1. Given a document set that need to be summarized, first of all, we split the document collection into sentences in such a way that each sentence is preceded by its corresponding document number and sentence position number. Next, SENNA semantic role labeler [30, 31] is employed to extract predicate argument structure from each sentence in the document set. The document set containing collection of predicate argument structures is modeled as weighted undirected graph (as described in Section 3.1.3) in a way that if the similarity weight $sim(p_i,p_j)$ between two predicate argument structures PASs p_i and p_j (i≠j) is greater than 0 then a link is established between them, otherwise no link is established. A modified graph based ranking algorithm (i.e. we incorporated PAS-PAS semantic similarity instead of content similarity measure into the ranking algorithm) is applied to determine salience (importance) score of the graph nodes (PASs) as discussed in Section 3.2.

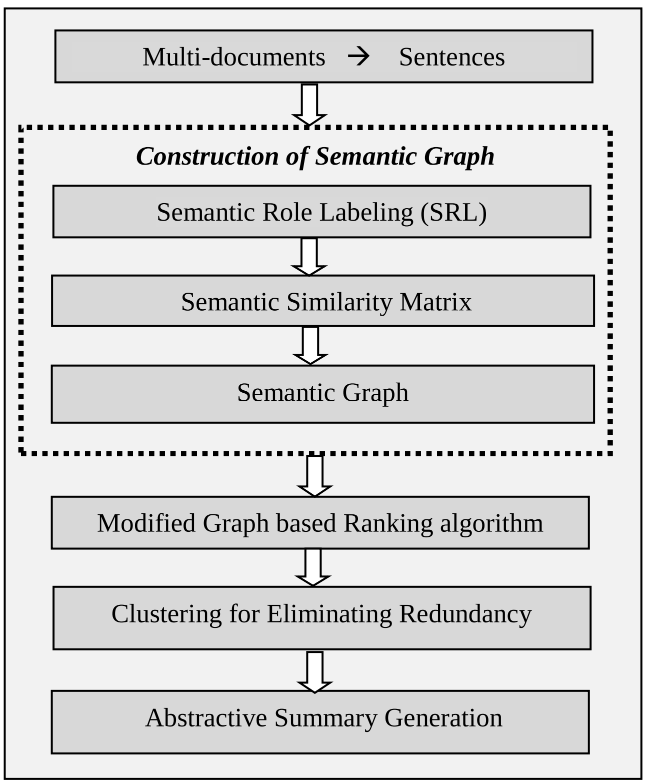


Figure 1 Proposed clustered semantic graph approach for multi-document abstractive summarization

The graph nodes are ranked based on salience score. In order to remove the redundant PASs, we perform agglomerative hierarchical clustering to group semantically similar PASs and choose the PAS with the highest salience score from each cluster as representative. The number of clusters are limited to the compression rate of summary. Finally, the selected graph nodes, which represent the predicate argument structures are fed to SimpleNLG realisation engine [32] to generate summary sentences.

3.1 Construction Of Semantic Graph

The goal of this phase is to construct semantic graph from the document set. At first, the document set is segmented into sentences and then semantic role labeling (SRL) is employed to the sentence collection to obtain their corresponding predicate argument structures (PASs). Next, we compute semantic similarity between each pair of predicate argument structure. Once the semantic similarity score for each pair of predicate argument structure is obtained, then semantic similarity matrix is constructed from the similarity scores of the predicate argument structures. We build semantic graph from semantic similarity matrix in such a way that the predicate argument structures (PASs) forms the vertices of the graph, and edge e_{ij} is associated with similarity weight $sim(p_i,p_j)$ between predicate argument structures p_i and p_j (i≠j). The similarity weight between predicate argument structures is determined based on Lin semantic similarity measure.

3.1.1 Semantic Role Labeling

The objective of Semantic Role Labeling (SRL) is to determine the syntactic constituents/arguments of a sentence with respect to the sentence predicates, identify the semantic roles of the arguments such as Agent, Patient, Instrument etc., and the adjunctive arguments of the predicate such as Locative, Temporal, Manner etc [33]. SRL has been extensively employed in text content analysis tasks such as text retrieval [34], information extraction [35], text categorization [36] and sentiment analysis [37].

As abstractive summarization requires deeper semantic analysis of text, therefore, this study employs semantic role labeling to extract predicate argument structure from source text in order to capture the semantics (meaning) of text (e.g. who did what to whom, when and how). The approach uses SENNA [30] tool for semantic role labeling (SRL), part-of-speech (POS) tags, and named entity recognition. SENNA is tested on the PropBank test set and achieved a per word error rate of approximately 14.5%, which is competitive with other state of the art methods [38].

At first, we split the document set into sentences in such a way that each sentence is preceded by its corresponding document number and sentence position number. Next, SENNA semantic role labeler is employed to parse each sentence and properly labels the semantic word phrases. These phrases are referred to as semantic arguments. The semantic arguments can be grouped in two categories: core arguments *(Arg)* and adjunctive arguments *(ArgM)* as shown in Table 1. In this study, we consider *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*.

Table 1 Core Arguments and Adjunctive 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

We consider all the complete predicate associated with the single sentence structure in order to avoid loss of important terms contributing to the meaning of sentence, and the actual predicate of the sentence.

We assume that predicates are complete if they have at least two semantic arguments. A sentence containing one predicate is represented by simple predicate argument structure, while a sentence containing more than one complete predicate is represented by a composite predicate argument structure.

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

The corresponding simple predicate argument structures P_1 and P_2 are obtained after applying semantic role labeling to sentences S_1 and S_2:

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]

3.1.2 Processing of Predicate Argument Structures

The predicate argument structures (PASs) once obtained are split into meaningful words or tokens. The stop words in PASs are removed and the remaining words are stemmed to their base form using porter stemming algorithm [39]. Next, SENNA POS tagger [30] is used to label each term of semantic arguments (associated with the predicates), with part of speech (POS) tags or grammatical roles. The POS tags NN stands for noun, V for verb, JJ for adjective and RB for adverb etc. In this study, we will compare predicates (Verbs) and the only terms of the semantic arguments of the predicates, which are labeled as noun (NN) and the rest are ignored. After POS tagging, the two predicate argument structures P_1 and P_2 in example 1 after processing are as follows:

P_1: [**A0**: a massive (JJ) hurricane NN] [**V**: attack] [**A1**: my home (NN)]

P_2: [AM-TMP: Eventually (RB)] [**A0**: a huge (JJ) cyclone (NN)] [**VBD**: hit] [**A1**: the entrance (NN) of my house (NN)]

This study compares predicate argument structures based on noun-noun, verb-verb, location-location and time-time arguments. Therefore, we extract only tokens from predicate argument structures, which are labeled as noun, verb, location, and time. All the PASs associated with the sentence will be included in comparison. Once the nouns, verbs, and other arguments (time and location) if exist, are extracted, the predicate argument structures obtained in example 1 after further processing will become

P_1: [A0: hurricane NN] [V:attack] [A1: home (NN)]

P_2: [AM-TMP: Eventually (RB)] [A0: cyclone (NN)] [VBD: hit] [A1: entrance (NN), house (NN)]

3.1.3 Semantic Similarity Matrix

The aim of this phase is to construct a matrix of semantic similarity scores for each pair of predicate argument structure. In this phase, similarity of the predicate argument structures (PASs) is computed pair wise based on acceptable comparisons of noun-noun, verb-verb, location-location and time-time. The comparisons of noun-noun, verb-verb in the respective PAS are accomplished with Lin semantic similarity measure, while the comparisons of location-location and time-time are achieved with edit distance algorithm.

Generally, semantic similarity measures have been applied in natural language processing (NLP) [40], word sense disambiguation [40], information retrieval [40, 41], question answering [42], recommender system [43], text segmentation [44], information extraction [43, 45] and so on. Most recently, the similarity measures that are based on WordNet have gained great significance by manifesting their strengths in making these applications more intuitive and intelligent. In this regard, several semantic similarity measures have been introduced and a comprehensive overview of these measures can be found in [46]. Based on experimental results in the literature [40], information content based measures lead the other measures and has the closest correlation with human judgment amongst all the semantic similarity measures. In this study, we utilize Lin measure semantic similarity measure [22].

Given a document set containing the collection of predicate argument structures (PASs), first we measure the similarity weight $sim(p_i, p_j)$ between predicate argument structures p_i and p_j based on Lin semantic similarity measure. In this step, at first the similarity of the predicate argument structures (PASs) is computed pair wise based on acceptable comparisons of noun-noun, verb-verb, location-location and time-time. The study exploits Lin semantic similarity measure for computing semantic similarity between each pair of PASs. Lin's measure is information content based measure and consider that each concept in the WordNet [47] hold certain information. This measure employs information content of the least common subsumer which states the shared information of the given two concepts, and the information content of the given concepts required to fully describe them.

$$sim_{Lin}(c_1, c_2) = \frac{2 \times IC\big(lso(c_1, c_2)\big)}{IC(c_1) + IC(c_2)} \quad (1)$$

First, Lin's measure uses WordNet to compute the least common subsumer (**lso**) of two concepts, which is the closest shared parent of the two concepts, then determines $IC(C1), IC(C2), and\ IC(lso\ (C1,\ C2))$. The information content (IC) of concept is achieved by determining the probability of occurrence of a concept in a large text corpus and quantified as follows:

$$IC(C) = -\log P(C) \quad (2)$$

Where $P(C)$ is the probability of occurrence of concept 'C' and is computed as follows:

$$P(C) = \frac{Freq(C)}{N} \quad (3)$$

Where Freq(C) is the number of occurrences of concept 'C' in the taxonomy and N is the maximum number of nouns.

Given two sentences $S_i\ and\ S_j$, the semantic similarity between their corresponding predicate argument structures p_i and p_j is represented by $sim_{sem}(p_i, p_j)$ and is determined using Eq. (8), where $sim_{verb}(p_i, p_j)$ is the similarity between predicates (verbs) determined using Eq. (5), $sim_{arg}(p_i, p_j)$ is the sum of similarities between the corresponding arguments of the predicates determined using Eq. (4). Both equations (4) and (5) exploit Lin's semantic similarity measure for computing similarity between noun terms in the semantic arguments of the predicate argument structures and the verbs of predicate argument structures respectively. Similarity between corresponding temporal arguments i.e. $sim_{tmp}(p_i, p_j)$ is computed using Eq. (6) and similarity between corresponding location arguments i.e. $sim_{loc}(p_i, p_j)$ is calculated using Eq. (7).
Since Lin's measure is based on WordNet, the temporal and location arguments may not be found in the WordNet, therefore we use edit distance algorithm in equations (6) and (7) for computing possible match/similarity between temporal and location arguments of the predicates. The similarity between the two predicate argument structures is computed using eq. (4-8).

$$sim_{arg}(p_i, p_j) = sim(A0_i, A0_j) + sim(A1_i, A1_j) + sim(A2_i, A2_j) \quad (4)$$

$$sim_{verb}(p_i, p_j) = \big(sim(Verb_i, Verb_j)\big) \quad (5)$$

$$sim_{tmp}(p_i, p_j) = \Big(sim\big(Tmp_i, Tmp_j\big)\Big) \quad (6)$$

$$sim_{loc}(p_i, p_j) = \big(sim(Loc_i, Loc_j)\big) \quad (7)$$

Equations (4), (5), (6), (7) are combined to give equation (8) as follows:

$$sim_{sem}(p_i, p_j) = sim_{verb}(p_i, p_j) + [sim_{arg}(p_i, p_j) + sim_{tmp}(p_i, p_j) + sim_{loc}(p_i, p_j) \quad (8)$$

Once the semantic similarity score for each pair of predicate argument structure is obtained, then semantic similarity matrix is constructed from the similarity scores of predicate argument structures.

3.1.4 Semantic Graph

The goal of this phase is to build semantic graph from the semantic similarity matrix constructed in previous phase. The undirected weighted semantic graph is constructed from similarity matrix (representing predicate argument structures (PASs) similarity scores) in such a way if the similarity weight $sim(p_i, p_j)$ between two predicate argument structures PASs p_i and p_j (i≠j) is greater than 0 then a link is established between them, otherwise no link is established. We let $sim(p_i, p_i) = 0$ to avoid self transitions. In this study, we are interested in only significant semantic similarities and eliminate the low values by defining similarity threshold that is empirically set to 0.5 [11]. So, a link is added between predicate argument structures (vertices) whose semantic similarity lies in the range of 0< β ≤ 0.5; otherwise no link is established.

Formally, given a document set D, let G=(*V*, *E*) is an undirected weighted graph that reveals the semantic relationship between predicate argument structures in the document set. Let Vs represents the set of vertices and each vertex v_i in Vs is the predicate argument structure in the document set. Let Es represents the set of edges and each edge e_{ij} in Es is labeled with the semantic similarity weight between predicate argument structures v_i and v_j (*i≠j*). The similarity weight between two predicate argument structures v_i and v_j is computed is computed using Eq. (8) and it is written formally as follows:

$$f(v_i, v_j) = sim_{sem}(v_i,\ v_j) \qquad (9)$$

3.2 Graph Based Ranking Algorithm

Conventionally, Google's PageRank [48] and HITS algorithm [49] are graph based ranking algorithms that have been effectively employed in Web-link analysis and social networks. The ranking algorithms have also been applied to text processing applications such as single and multi-document extractive summarization [11]. PageRank algorithm [48] applied on undirected graph achieved the best performance in DUC 2002 multi-document extractive summarization task. The PageRank procedure provides the means for determining the significance of a vertex within a graph, by considering global information from the whole graph.

Previous graph based methods exploit relationships/associations between sentences based on content similarity rather than semantic similarity, and apply similar procedure like PageRank to choose sentences based on number of "votes", received from their neighbouring sentences. To our knowledge, the graph based ranking algorithm has not been considered for multi-document abstractive summarization, and this study will employ a modified page rank which will take into account the PAS-PAS semantic relationship in the PAS ranking process (or importance analysis).

We let denote the salience or importance score of predicate argument structure v_i by $PAS_{Score}(v_i)$ and it can be inferred from all those predicate argument structures that are connected to it; and we formulate it in a recursive manner as in the PageRank algorithm as follows:

$$PAS_{Score}(v_i) = \sum_{all\ j \neq i} PAS_{Score}(v_j).\ \widehat{M}_{j,i} + \frac{(1-\mu)}{|V|} \qquad (10)$$

The corresponding matrix form is

$$\vec{\lambda} = \mu\, \widehat{M}^T \vec{\lambda} + \frac{(1-\mu)}{|V|}\, \vec{e} \qquad (11)$$

Where $\vec{\lambda} = \left[PAS_{Score}(v_j)\right]_{|V|\times 1}$ is the vector of PAS salience scores. $\vec{e}$ is a vector in which all elements equal to 1. μ denotes the damping factor in the PageRank algorithm, and usually assigned a value of 0.85.

The above procedure can be assumed as a Markov chain by considering predicate argument structures as states of the chain and for the given matrix in Eq. (11), the corresponding transition matrix is obtained as follows:

$$Z = \mu\, \widehat{M}^T + \frac{(1-\mu)}{|V|}\, \vec{e}\, \vec{e}^T \qquad (12)$$

The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix given in Eq. (12). From implementation perspective, the initial scores of all vertices (PASs) are set to 1 and the iteration/ranking algorithm employs Eq. (10), which is based on PageRank. The iteration algorithm is run on undirected weighted graph to calculate the new salience/ranking scores of the vertices (PASs). The iteration algorithm keeps on computing the salience scores of the vertices until convergence is achieved. The converge is achieved by the iteration/ranking algorithm, when the difference between the ranking scores determined for any vertices (PASs) at two successive iterations falls below a given threshold (0.0001 in this work) [11]. After the convergence is achieved, the ranking scores obtained for vertices (PASs) of the graph are sorted in reverse order.

3.3 Clustering To Remove Redundant Predicate Argument Structures

Clustering of sentences for the purpose of removing redundancy is a common step in multi-document summarization. Agglomerative hierarchical clustering is well-known technique in the hierarchical clustering method, which has been found useful in the range of applications [50]. There are five well-known linkage methods of agglomerative hierarchical clustering (HAC) [51] i.e. single linkage, complete linkage, average linkage, ward and centroid method. Based on different measures (Entropy and F-Score and Kendall W test), it was found from the literature studies in [52], [53] and [54] that average linkage is the most suitable method for document clustering. Therefore, this study exploits HAC algorithm based on average linkage method. At first the semantic graph is represented as semantic similarity matrix and given as input to the HAC algorithm. We consider the value at position (i,j) in the semantic similarity matrix as semantic similarity between *i^th^ and j^th^* clusters, assuming that the construction of similarity matrix begins with each predicate argument structure as a single cluster. The pseudo code for clustering similar predicate argument structures is given below.

Pseudo code for Agglomerative Clustering Algorithm

Input: Semantic Similarity Matrix
Output: Clusters of similar predicate argument structures

a. Merge the two clusters that are most similar.
b. Update the semantic similarity matrix to represent the pair wise similarity between the newest cluster and the original cluster based on average linkage method
c. Repeat step 1 and 2 until the compression rate of summary is reached

In this study, we consider 20% compression rate of summary. Once the clusters are obtained, predicate argument structure with the highest salience score from each cluster is chosen as the most representative. Finally, we arrange the chosen representative predicate argument structures from each cluster based on salience scores in descending order.

3.4 Summary Generation

This phase takes the top scored representative argument structures (PASs) from previous phase, employs SimpleNLG [32] and a simple heuristic rule implemented in SimpleNLG, to generate summary sentences from PASs.

The simple 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 such as "and".

SimpleNLG is an English realisation engine which provides simple interfaces to produce syntactical structures and transform them into sentences using simple grammar rules. Moreover, the significant advantage of this engine is its robustness i.e. the engine will not crash when the input syntactical structures are incomplete or ill-formed.

As discussed in section 3.1.1, we consider 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 the rest of the arguments are ignored. Thus, the final summary sentences generated from the given predicate argument structures will be the compressed version of the original source sentence in most cases. During summary sentence generation process through SimpleNLG, the simple heuristic rule implemented in SimpleNLG combines the predicate argument structures that refer to the same subject (entity). The following example demonstrates how we generate abstractive summary from the given source input sentences.

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]

We assume that P_1, P_2 and P_3 are the top scored predicate argument structures selected in previous step. According to the rule stated above, the subject *A0* is identified as repeated in the above example and is eliminated from all predicate argument structures except the first one. The SimpleNLG applies the heuristic rule on the above three predicate argument structure and form the summary sentence that is compression version of the original source sentences.

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.

4.0 EVALUATION RESULTS

The proposed semantic graph based approach for multi-document summarization is evaluated using DUC 2002 document sets (DUC, 2002), which is a standard corpus used in text summarization research, and contains documents along with their human model summaries (both extractive and abstractive summaries). DUC 2002 data set contains 59 document sets produced by the National Institute of Standards and Technology (NIST). The data set chosen for our work refers to task2 and task3 defined for the data set. There are also more recent editions of DUC data sets, however they lack human produced abstracts and deal with other summarization tasks such as focused summarization, question- answering, update summarization etc.

The two standard evaluation metrics, Recall-Oriented-Understudy for Gisting Evaluation (ROUGE) [55] and Pyramid [56] have been widely used in the context of evaluation of text summary. Previous research studies showed that ROUGE metric has been employed for the evaluation of extractive summaries. ROUGE score is the n-gram exact matches between system summary and human model summaries. Another evaluation metric, called the Pyramid metric is used for the evaluation of abstractive summaries. The obvious advantage of Pyramid metric over the ROUGE is that it can capture different sentences in the summaries that uses different words but express similar meanings [56].

This study employs both Pyramid and ROUGE evaluation metrics for the evaluation of our proposed approach. We employ Pyramid evaluation results to compare our proposed semantic approach (Sem-Graph) with the recent abstractive approach for multi-document summarization (AS) [16], the best system, average of automatic systems, and average of human model summaries, in the context of multi-document abstractive summarization shared task in DUC 2002.

On other hand, we employ ROUGE-1 and ROUGE-2 evaluation measures to compare our approach (Sem-Graph) with the recent graph based multi-document extractive summarization approach (Event graph) [29], best system, and average of human model summaries, in the context of DUC 2002 multi-document extractive summarization shared task. Primarily, ROUGE-1 and ROUGE-2 measures determine recall for by comparing system summaries against human produced summaries (extracts).

The Pyramid metric measures the quality of system generated summary by comparing it with human model summaries(abstracts). Pyramid score (Mean Coverage Score) [56] for peer summary or candidate summary is computed as follows.

$$Mean\ Coverge\ Score\ = \frac{Total\ Peer\ SCUs\ Weight}{Average\ SCU\ in\ the\ Model\ Summary} \quad (13)$$

Where SCUs refers to the summary content units and their weights correspond to number of model (human) summaries they appeared in.

The precision for peer summary [56] or candidate summary is computed as follows.

$$Precision\ = \frac{Number\ of\ Model\ SCUs\ \ in\ Peer\ Summary}{Average\ SCU\ in\ the\ Peer\ Summary} \quad (14)$$

The F-measure for peer summary can be computed from equations (13) and (14) as follows:

$$F-Measure = \frac{2 \times Mean\ Coverage\ Score \times Precision}{Mean\ Coverage\ Score + Precision} \quad (15)$$

There are many variants of ROUGE evaluation measures, however, it is confirmed from the literature that ROUGE-1, ROUGE-2 are effectively employed for multi-document extractive summarization [55]. $\mathrm{ROUGE-N}$ can be defined [55] as is an n-gram recall between a system summary and a set of reference summaries, and is calculated as follows

$$ROUGE-N = \frac{\sum_{S\in\{Reference\ Summaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S\in\{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

Where n is the length of the *n-gram*, $gram_n$ and $count_{match}\ (gram_n)$ is the maximum number of n-grams that simultaneously occur in a system summary and a set of human summaries.

For each data set amongst the 59 news articles/data, our proposed approach generates a 100 words summary, the tasks undertaken by other systems participating in multi-document abstractive and extractive summarization tasks. To compare the performance of our proposed approach (Sem-Graph) in the context of DUC 2002 multi-document abstractive summarization shared task, we setup three comparison models (Best, Avg, AS), besides the average of human model summaries (Models). For comparative evaluation, Table 2 shows comparison of abstractive summarization results for different systems over the mean coverage score, average precision and average F-measure obtained on DUC 2002 dataset. Figure 2 above visualizes the summarization results obtained with the proposed approach and other comparison models.

Table 2 Comparison of multi-document abstractive summarization results in DUC 2002 based on mean coverage score, average precision and average F-Measure

System	Mean Coverage Score	Avg-Precision	AVG-F-Measure
Models	0.6910	0.8528	0.7634
Best (System 19)	0.2783	0.7452	0.4053
Avg	0.1775	0.6700	0.2806
AS [16]	0.4378	0.643	0.5209
Sem-Graph	0.5141	0.7512	0.6104

Best (System 19) is the best abstractive summarization system in DUC 2002, Avg denotes the average of abstractive summarization systems participating in DUC 2002, **AS** denotes the recent abstractive approach for multi-document summarization and the Models denote the average of human model abstractive summaries.

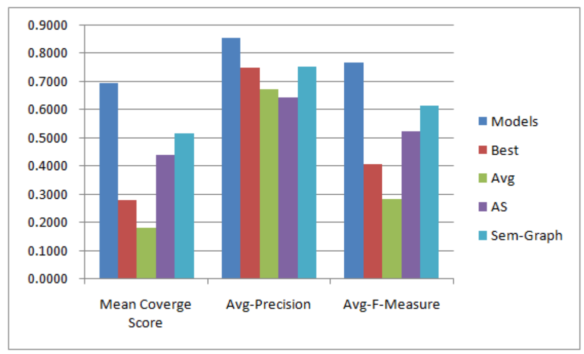


Figure 2 Comparison of summarization results based on mean coverage score, average precision and average f-measure

To compare the performance of our proposed approach (Sem-Graph) in the context of DUC 2002 multi-document extractive summarization shared task, we setup two comparison models (Best, Event graph), besides the average of human model summaries (Models). Event graph refers to the recent graph based approach for multi-document extractive summarization.

Table 3 shows comparative evaluations of the proposed approach with other extractive summarization models based on Recall obtained with ROUGE-1 and ROUGE-2 measures, achieved on DUC 2002 data set.

Table 3 Comparison of the proposed multi-document abstractive summarization system (Sem-Graph) with multi-document extractive summarization systems in DUC 2002 based on recall obtained with ROUGE-1 and ROUGE-2 measures

System	ROUGE-1 Recall	ROUGE-2 Recall
Best (System 21)	0.395	0.103
Models	0.418	0.102
Event graph [29]	0.415	0.116
Sem-Graph	0.40	0.099

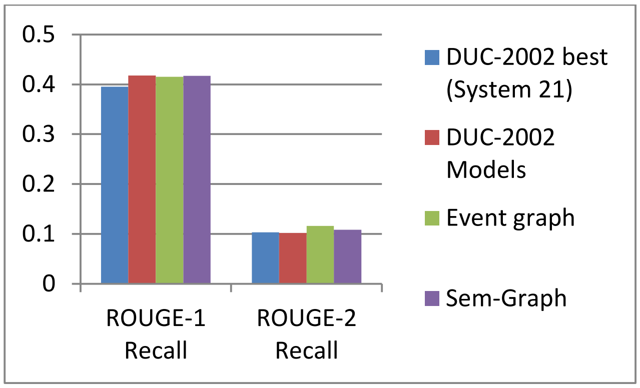


Figure 3 Comparison of the proposed multi-document abstractive summarizations system (Sem-Graph) with multi-document extractive summarization systems in DUC 2002 based on recall obtained with ROUGE-1and ROUGE-2

5.0 DISCUSSION

This section discusses the results presented in previous section. Table 2 presents the comparative evaluation of the proposed approach and other abstractive summarization systems based on Pyramid evaluation measures (mean coverage score, average precision and average F-measure), in the context of DUC 2002 multi-document abstractive summarization shared task. On other hand, Table 3 demonstrates the comparative evaluation of the proposed approach with extractive summarization systems based on recall obtained with ROUGE-1 and ROUGE-2 measures, in the respective DUC 2002 multi-document extractive summarization shared task.

First we discuss the results given in Table 2, we can observe that on mean converge score, average precision and average F-measure, the proposed approach (Sem-Graph) outperforms the best system and the recent multi-document abstractive summarization approach (AS) [16], and came second to the average of human model summaries (Models). In order to validate the results of the proposed approach and other comparison models, we also carried out a statistical significance test (Paired-Samples T-test), and achieved low significance value of $p < 0.05$. These results suggest that the summary produced by our approach (Sem-Graph) is more closer to the way humans produce summary as compared to other comparison models (Best, Avg, AS).

Moreover, refer to the results given in Table 3, the proposed summarization approach (Sem-Graph) performs better than the best system and came third to the human models based on ROUGE-1 recall. However, based on ROUGE-2 recall, the performance of the proposed approach slightly degrades as compared to other comparison models. This might be due to the fact that our proposed abstractive summarization approach generates summary that contains compressed version of original source sentences; while on other hand, extractive summarization systems generate summary that contains original source sentences. ROUGE-1 and ROUGE-2 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 less matching text snippets with the human produced summary as compared to the other extractive summarization systems.

Paired-Samples T-test is also carried out to validate the results of the proposed approach and other extractive summarization models and obtained a significance value of $p < 0.05$.

We can observe that the PAS to PAS semantic similarity employed in the proposed approach assists in detecting redundancy by capturing semantically equivalent PASs. Thus the summarization results are improved by eliminating redundant PASs from the perspective of Pyramid measures. However, the summarization results of our approach are closer to benchmark systems from the perspective of ROUGE-1 and ROUGE-2 measures.

6.0 CONCLUSION

The proposed semantic graph based approach shows the feasibility of new direction towards abstractive summarization research. We believe that the proposed work aims at the real goals of automatic summarization – controlling the content and structure of the summary. The proposed clustered semantic graph based approach is evaluated in the context of DUC 2002 multi-document summarization shared tasks. One task refers to abstractive summarization while the other task refers to extractive summarization. The approach assumes semantic structure of sentence - predicate argument structure, automatically extracted by employing semantic role labeling as opposed to other graph based extractive approaches, most of which consider sentence as bag of words. Our approach exploits Lin semantic similarity measure to detect redundancy by capturing semantically equivalent predicate argument structures. On other hand, existing graph based approaches cannot capture redundant sentences that are semantically equivalent as they mostly rely on cosine similarity measure.

Moreover, the approach is promising enough to be applicable to any domain and does not require any intervention of human experts. In future, we will explore Cross-Document Structural Theory (CST) relations for multi-document abstractive summarization and examine their impact on summarization.

Acknowledgement

This work is supported by the Higher Education Commission (HEC), Islamia College Peshawar (Chartered University), Pakistan and Soft Computing Research Group (SCRG) of Universiti Teknologi Malaysia (UTM). This Work is also supported in part by grant from Vot 07H89.

References

[1] M. A. Fattah and F. Ren. 2009. GA, MR, FFNN, PNN and GMM based Models for Automatic Text Summarization. *Computer Speech & Language*. 23: 126-144.
[2] R. Barzilay and K. R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*. 31: 297-328.
[3] D. Das and A. F. Martins. 2007. A Survey on Automatic Text Summarization. *Literature Survey for the Language and Statistics II course at CMU*. 4: 192-195.
[4] S. Ye, T.-S. Chua, M.-Y. Kan, and L. Qiu. 2007. Document Concept Lattice for Text Understanding and Summarization. *Information Processing & Management*. 43: 1643-1662.
[5] J. Kupiec, J. Pedersen, and F. Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th annual International ACM SIGIR Conference on Research and Development In Information Retrieval*. 68-73.
[6] K. Knight and D. Marcu. 2000. Statistics-based Summarization-Step One: Sentence Compression. In *Proceedings of the National Conference on Artificial Intelligence*. 703-710.
[7] B. Larsen. 1999. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. *Advances in Automatic Text Summarization*. 71.
[8] M. A. Fattah. 2014. A Hybrid Machine Learning Model for Multi-Document Summarization. *Applied Intelligence*. 40: 592-600.
[9] G. Erkan and D. R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summarization. In *EMNLP*. 365-371.
[10] G. Erkan and D. R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res.(JAIR)*. 22: 457-479.
[11] R. Mihalcea and P. Tarau. 2005. A Language Independent Algorithm for Single and Multiple Document Summarization.
[12] X. Wan and J. Yang. 2006. Improved affinity graph based multi-document summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. 181-184.
[13] L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. The PageRank Citation Ranking: Bringing Order to the Web.
[14] R. Barzilay, K. R. McKeown, and M. Elhadad. 1999. 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*. 550-557.
[15] H. Tanaka, A. Kinoshita, T. Kobayakawa, T. Kumano, and N. Kato. 2009. Syntax-driven Sentence Revision for Broadcast News Summarization. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*. 39-47.
[16] P.-E. Genest and G. Lapalme. 2011. Framework for Abstractive Summarization Using Text-To-Text Generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. 64-73.
[17] S. M. Harabagiu and F. Lacatusu. 2002. Generating Single and Multi-Document Summaries With Gistexter. In *Document Understanding Conferences*.
[18] P.-E. Genest and G. Lapalme. 2012. Fully Abstractive Approach to Guided Summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. 354-358.
[19] C.-S. Lee, Z.-W. Jian, and L.-K. Huang. 2005. A Fuzzy Ontology and Its Application to News Summarization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*. 35: 859-880.
[20] C. F. Greenbacker. 2011. Towards a Framework for Abstractive Summarization of Multimodal Documents. *ACL HLT 2011*. 75.
[21] I. F. Moawad and M. Aref. 2012. Semantic Graph Reduction Approach for Abstractive Text Summarization. In *Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on*. 132-138.
[22] D. Lin. 1998. An Information-theoretic Definition of Similarity. UIn *ICML*. 296-304.
[23] J. C. K. Cheung and G. Penn. 2013. Towards Robust Abstractive Multi-Document Summarization: A Caseframe Analysis of Centrality and Domain. In *ACL (1)*. 1233-1242.
[24] I. Mani and E. Bloedorn. 1999. Summarizing Similarities and Differences Among Related Documents. *Information Retrieval*. 1: 35-67.
[25] J. Zhang, L. Sun, and Q. Zhou. 2005. A Cue-based Hub-Authority Approach for Multi-Document Text Summarization. In *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*. 642-645.
[26] F. Wei, W. Li, Q. Lu, and Y. He. 2010. A Document-Sensitive Graph Model for Multi-Document Summarization. *Knowledge and Information Systems*. 22: 245-259.
[27] S. S. Ge, Z. Zhang, and H. He. 2011. Weighted Graph Model Based Sentence Clustering and Ranking for Document Summarization. In *Interaction Sciences (ICIS). 2011 4th International Conference on*. 90-95.
[28] T.-A. Nguyen-Hoang, K. Nguyen, and Q.-V. Tran. 2012. TSGVi: A Graph-Based Summarization System for Vietnamese Documents. *Journal of Ambient Intelligence and Humanized Computing*. 3: 305-313.
[29] G. Glavaš and J. Šnajder. 2014. Event Graphs for Information Retrieval and Multi-Document Summarization. *Expert Systems with Applications*. 41: 6904-6916.
[30] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *The Journal of Machine Learning Research*. 12: 2493-2537.
[31] T. Barnickel, J. Weston, R. Collobert, H.-W. Mewes, and V. Stümpflen. 2009. Large Scale Application of Neural Network Based Semantic Role Labeling for Automated Relation Extraction from Biomedical Texts. *PLoS One*. 4: e6393.
[32] A. Gatt and E. Reiter. 2009. SimpleNLG: A Realisation Engine for Practical Applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*. 90-93.
[33] C. Aksoy, A. Bugdayci, T. Gur, I. Uysal, and F. Can. 2009. Semantic Argument Frequency-based Multi-Document Summarization. In *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*. 460-464.
[34] S. Shehata, F. Karray, and M. S. Kamel. 2013. An Efficient Concept-based Retrieval Model for Enhancing Text Retrieval Quality. *Knowledge and Information Systems*. 35: 411-434.

[35] L. Del Corro and R. Gemulla. 2013. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd International Conference on World Wide Web*. 355-366.
[36] J. Persson, R. Johansson, and P. Nugues. 2008. Text Categorization Using Predicate–Argument Structures. *Proe. the*. 1: 142-149.
[37] N. Jadhav and P. Bhattacharyya. 2014. Dive Deeper: Deep Semantics for Sentiment Analysis. *ACL 2014.* 113.
[38] S. S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky. 2004. Shallow Semantic Parsing using Support Vector Machines. In *HLT-NAACL*. 233-240.
[39] M. F. Porter. 2001. Snowball: A Language for Stemming Algorithms. ed.
[40] Y. Li, Z. A. Bandar, and D. McLean. 2003. An Approach for Measuring Semantic Similarity Between Words Using Multiple Information Sources. *Knowledge and Data Engineering, IEEE Transactions on.* 15: 871-882.
[41] R. Rada, H. Mili, E. Bicknell, and M. Blettner. 1989. Development and Application of a Metric on Semantic Nets. *Systems, Man and Cybernetics, IEEE Transactions on.* 19: 17-30.
[42] A. G. Tapeh and M. Rahgozar. 2008. A Knowledge-based Question Answering System for B2C Ecommerce. *Knowl.-Based Syst.* 21: 321-326.
[43] Y. Blanco-Fernández, J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, J. García-Duque*, et al.* 2008. A Flexible Semantic Inference Methodology to Reason About User Preferences in Knowledge-Based Recommender Systems. *Knowledge-Based Systems*. 21: 305-320.
[44] H. Kozima. 1994. Computing Lexical Cohesion as a Tool for Text Analysis. Citeseer.
[45] M. Stevenson and M. A. Greenwood. 2005. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. 379-386.
[46] L. Meng, R. Huang, and J. Gu. 2013. A Review of Semantic Similarity Measures in Wordnet. *International Journal of Hybrid Information Technology.* 6: 1-12.
[47] G. A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*. 38: 39-41.
[48] S. Brin and L. Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems.* 30: 107-117.
[49] J. M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM).* 46: 604-632.
[50] F. Murtagh and P. Contreras. 2011. Methods of Hierarchical Clustering. *arXiv preprint arXiv:1105.0121*.
[51] S. Takumi and S. Miyamoto. 2012. Top-down vs Bottom-Up Methods of Linkage for Asymmetric Agglomerative Hierarchical Clustering. In *Granular Computing (GrC), 2012 IEEE International Conference on*. 459-464.
[52] M. Steinbach, G. Karypis, and V. Kumar. 2000. A Comparison of Document Clustering Techniques. In *KDD workshop on Text Mining*. 525-526.
[53] Y. Zhao, G. Karypis, and U. Fayyad. 2005. Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery.* 10: 141-168.
[54] A. El-Hamdouchi and P. Willett. 1989. Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval. *The Computer Journal.* 32: 220-227.
[55] C.-Y. Lin. 2004. Rouge: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 74-81.
[56] A. Nenkova and R. Passonneau. 2004. Evaluating Content Selection in Summarization: The Pyramid Method.