Applied Soft Computing 21 (2014) 265–279

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi document summarization based on news components using fuzzy cross-document relations

Yogan Jaya Kumar [a,b,*], Naomie Salim [b], Albaraa Abuobieda [c], Ameer Tawfik Albaham [b]

[a] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia
[b] Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[c] Faculty of Computer Studies, International University of Africa, 2469 Khartoum, Sudan

ARTICLE INFO

Article history:
Received 4 February 2013
Received in revised form 27 January 2014
Accepted 30 March 2014
Available online 12 April 2014

Keywords:
Multi document summarization
News components
Cross-document structure theory (CST)
Case-based reasoning
Genetic algorithm
Fuzzy logic

ABSTRACT

Online information is growing enormously day by day with the blessing of World Wide Web. Search engines often provide users with abundant collection of articles; in particular, news articles which are retrieved from different news sources reporting on the same event. In this work, we aim to produce high quality multi document news summaries by taking into account the generic components of a news story within a specific domain. We also present an effective method, named Genetic-Case Base Reasoning, to identify cross-document relations from un-annotated texts. Following that, we propose a new sentence scoring model based on fuzzy reasoning over the identified cross-document relations. The experimental findings show that the proposed approach performed better that the conventional graph based and cluster based approach.

1. Introduction

The work on automatic text summarization can be dated back to the late 1950s [1]. Since then, the field of text summarization has witnessed continuous involvement by many researchers in the attempt to look for different strategies to automate text summarization [2–4]. The common goal of automatic text summarization is analogous to the reason why humans create summaries from text; i.e. to present a concise version of the original text to the reader. The need for automatic text summarization is even deemed necessary in the current Internet age.

With the fast growing of World Wide Web (WWW), access to online information had lead to the problem of information overload. Online news surfing provides readers with many articles since it involves multiple news sources. Google News, Columbia NewsBlaster and News In Essence are some of the popular online based news clustering systems that were built to alleviate information overload faced by netizens [5]. As aforementioned, Newsblaster, which is a fully deployed online news system, was built to summarize news from the Web [6]. The system identifies news stories through Web crawling and clusters them to specific topics.

One of the most common methods used in text summarization field is the feature based method. In the process of identifying important sentences, features influencing the relevance of sentences are determined. Some features that are often considered for sentence selection are word frequency, title words, cue words, sentence location and sentence length [2]. These features often increase the candidacy of a sentence for inclusion in summary. Feature based method are however commonly used for single document summarization. In the case of multi document summarization, two mainstream methods often employed are the cluster based method and the graph based method [7].

In this study, the aim of our research is to produce high quality multi document news summaries by taking into account the generic components of a news story; such as *who*, *what*, *when*, *where* and *how*. We believe that providing such contextual information coverage would be ideal for news summary creation. In this work, we deal with news stories related to natural disaster events; for example earthquakes, hurricanes, floods and others.

Since we are dealing with news stories, news documents which are related to the same topic usually contain semantically related textual units. This has motivated us to further investigate the utility of cross-document relations for identifying highly relevant sentences to be included in the summary. Radev [8] has initially proposed the idea of cross-document relations that exist

* Corresponding author at: Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia.
Tel.: +60 133915420.
E-mail address: yogan@utem.edu.my (Y.J. Kumar).

http://dx.doi.org/10.1016/j.asoc.2014.03.041
1568-4946/

in topically related documents. He came up with CST (Cross-document Structure Theory) model which describes the semantic relations between textual units such as words, phrases or sentences. In this study, we consider the semantic relations between sentences, for example, the relation between two sentences can be *Identity*, *Overlap*, *Description*, and etc. Complete descriptions on the CST relationship types can be found in Ref. [8].

The need for automatic identification of cross-document relation is indeed necessary for task related to multi document summarization. It is not efficient to be dependent on humans to perform such task for an automated system. Therefore, in this work, we will identify the cross-document relations from un-annotated documents by incorporating a novel integration of the genetic learning algorithm and the case base reasoning (CBR) model that is tailored to the task of classification. Following that, we propose a new sentence scoring model based on fuzzy reasoning over the identified cross-document relations. Then, based the fuzzy scoring, top ranking sentences will be selected to produce the multi document summary. Details on the overall architecture will be discussed later in Section 3.

The rest of this paper is organized as follows: Section 2 presents related works to this study. Section 3 outlines the overview of the approach together with the extraction process of the news component sentences. The cross-document relation identification using the Genetic-CBR approach is given in Section 4, while the fuzzy reasoning implementation is given in Section 5. Section 6 describes the experimental setting and results. We finally end with conclusions in Section 7.

2. Related work

A number of research works have addressed multi document summarization in academia and illustrated different types of approaches for multi document summarization [4,9]. However, there are two methods which are relatively common in multi document summarization studies; namely the cluster based method and graph based method [2,7].

The cluster based method which was pioneered by Radev et al. [10], uses cluster centroids, i.e. top ranking *tf-idf* (term frequency-inverse document frequency) to represent the clusters. Sentences from each cluster that are most similar to these centroids are then selected to be included in the summary. A widely used clustering algorithm is the *k*-means algorithm which is based on partitional clustering. Cluster based methods has been successful in its task to represent diversity and reduce redundancy within multiple articles. Some of the works that take the benefits of clustering approach to produce summary can be found in Refs. [11–14].

For the graph based method, its fundamental theory is supported by the links that exist between sentences. These links exist based on some measured similarity between the sentences. As in most literature concerning graph based approach, the commonly used similarity measure is the cosine similarity measure [15]. Sentences with high similarity weights (with respect to other sentences in the documents) will be ranked top for summary sentence selection. A well known graph based ranking algorithm is Google's PageRank [16] which has been traditionally used in Web-link analysis and social networks. The graph based method became popular for multi document summarization task as it was able to identify prestigious sentences across the documents [17–20]. However, if we look at the underlying concept of the graph based approach, the 'relation' between sentences is determined based only on its measured similarity value and not based on its relationship type.

As stated earlier in Section 1, the CST model defines the cross-document relations that exist between topically related documents. Following this, a number of researchers have addressed the benefits of CST for summarization task. In the work presented by Zhang et al. [21], they replace low-salience sentences with sentences that maximize total number of CST relations in the final summary. To discover the CST relations in a set of documents, they conducted experiments, in which human subjects were asked to find these relations over a multi-document news cluster. Similarly, Jorge and Pardo [22] worked on CST relations for content selection methods to produce preference-based summaries. They run their experiments with Brazilian Portuguese news texts (previously annotated with CST relations by human experts) where they rank sentence according to the number of CST relations it holds.

However the major limitation of the above works is that the CST relations need to be manually annotated by human experts; which is a drawback for an automatic summarization system. Our work, in contrast, treats this limitation by identifying the relations between sentences directly from un-annotated documents. Moreover, our fuzzy reasoning model is designed to rank sentence based on the type of relation it holds and not solely on the total number of relations.

Although there have been some attempts to learn the CST relations in texts, to our knowledge, only two interrelated works: [23] and [24] were evaluated on English texts, where the authors applied boosting classification algorithm to identify the presence of CST relations between sentences. However their classifier showed poor performance in classifying most of the CST relations; obtaining average values of 45% precision, 31% recall, and 35% *f*-measure. Zahri and Fukumoto [25] also attempted to identify some CST relationship types; however they did not report any evaluation. Besides English texts, CST parsing had also been studied for Brazilian Portuguese texts [26] and Japanese texts [27]. The authors of [26] experimented with three types of classifiers namely, the multi-class, hierarchical, and binary classifiers; and obtained a general accuracy of 41.58%, 61.50% and 70.51% respectively on unbalanced data. For multi-class classifiers, SVM was found to outperform other classifiers such as decision tree and Naïve Bayes. The authors of Ref. [27] however attempted only two relations, i.e. *Equivalence* and *Transition* and obtained an *f*-measure of 75.50% and 45.64% respectively using a SVM classifier.

In this study, we propose a supervised learning method based on case based reasoning (CBR) technique which is optimized using genetic learning algorithm for automatic cross-document relationships identification. Our general hypothesis for this study is that the generic news components integrated in news articles and the knowledge obtained from the automatically identified cross-document relations helps to improve the quality of multi document summary.

3. Overview of approach

In this section, we present the overall architecture of our proposed approach, i.e. multi document summarization based on news components using fuzzy cross-document relations. As highlighted in Fig. 1, there are three main phases; which includes component sentence extraction, cross-document relation (CST relation) identification and sentence scoring using fuzzy reasoning. Sections 3.1 and 3.2 will describe the generic components of news documents and its extraction process while Sections 4 and 5 will describe the Genetic-CBR and fuzzy reasoning implementations. For a given document set (in this case the news articles), the summarization steps for generating summary can be described in Algorithm 1.

Algorithm 1. Multi Document Summarization based on News Components Using Fuzzy Cross-Document Relations

1. Input document set D: take the document set D as input, $D = \{D_1, D_2, D_3, \ldots, D_n\}$

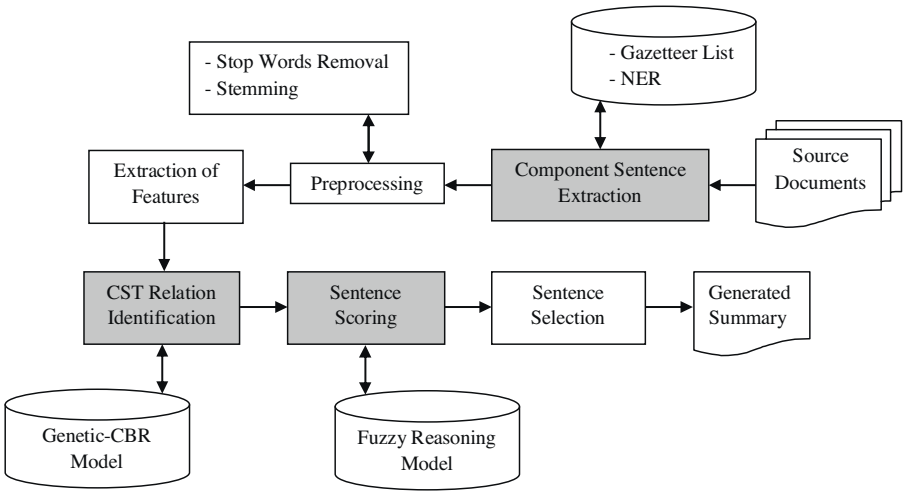

Fig. 1. General architecture of the proposed approach.

2. Component Sentence Extraction:
   2.1 Extract the component sentences using the gazetteer list and named entity recognition (see details in Section 3.2).
   2.2 Group each extracted sentence into its corresponding component cluster.
3. *Preprocessing*: perform stop word removal and word stemming on the sentences.
4. *Features extraction*: extract the features from each sentence pair, there are altogether five features, $F = \{CS, WO, LT, NP, VP\}$ (see details in Section 4).
5. *CST relation identification*: use the Genetic-CBR model to identify the CST relations for each sentence pair (refer to Algorithm 2).
6. *Sentence scoring*: use the fuzzy reasoning model to score the sentences (refer to Algorithm 3).
7. Repeat steps 4–6 for each component cluster.
8. Sentence selection:
   8.1 Remove redundant sentences from each component cluster using word overlap check.
   8.2 Select high ranking sentences from each component cluster based on the cluster size ratio until the desired summary length is met.
9. *Summary generation*: the final summary is obtained.

3.1. Generic components of news documents

As far as news documents are concerned, different news sources reporting on a particular event tend to contain common components that make up the main story of the news. The most common components of a news article consist of *who*, *what*, *when*, *where* and *how*. In the process of news story production, these are the core components which a journalist must collect, interpret, organize, and transmit [28]. Furthermore, such components are very close to how human perceive news contents.

Now, if we look into the context of natural disaster events (our working domain), its news story can be associated to components such as the description of the disaster (*how*), information about affected locations (*where*), persons involved (*who*), the damages to human and properties (*what*), the relief efforts (*what*) and the organizations involved (*who*); refer to Fig. 2. Such occurrence of component sentences with its contextual information content is what the readers usually look for while reading news stories (natural disaster events in this case).

In this work, we aim to incorporate such news components into the summarization process to produce better summaries. It should be noted that the component *when* was excluded since the summary produced here are for multiple news stories which were reported for the same event i.e. the event happens over the same time period. The following section will describe the extraction process of component sentences from the news documents.

3.2. Component sentence extraction

Over the years, a number of information extraction (IE) techniques have been developed. A comprehensive review and analysis of these techniques can be found in [29]. In this work, we have employed two techniques, namely named entity recognition (NER) and gazetteer lists, to extract the component sentences from the news articles. NER is generally used to identify entities that correspond to names of persons, locations and organizations. While the latter technique i.e. gazetteer lists can be used to identify other entities of interest. In fact, many IE systems have demonstrated the flexibility and effectiveness of this technique in various applications [30].

The entities or terms to be identified for a particular category are first placed in a list, known as the gazetteer list. In our work, these categories refer to the components we previously defined. For instance, the component *human damages* contains terms such as *dead*, *injured* and *missing* while the component *relief efforts* contains terms such as *aid*, *rescue* and *fund*. These component terms

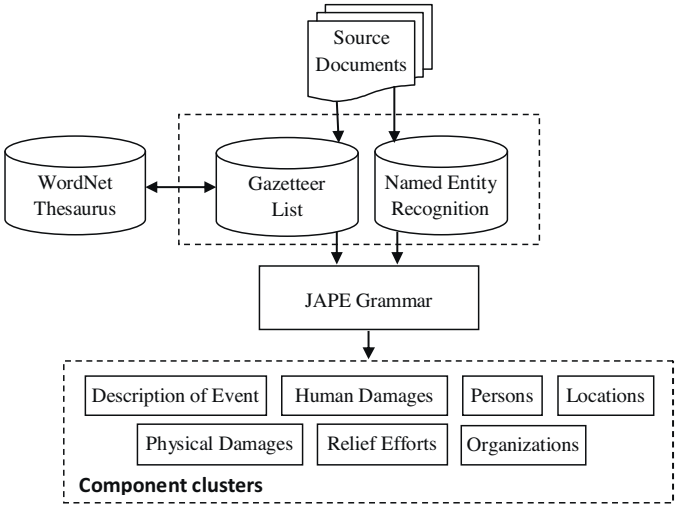

Fig. 2. Component sentence extraction.

```
Phase: COMPONENTSENTENCE
Input: Token Lookup Human_Damage Physical_Damage Relief _Effort
Event_Description Sentence
Options: control = appelt debug = true

Rule: Number1
(
({Sentence,Sentence contains Human_Damage }):sentence
)
-->
:sentence. Human_Damage_Sentence={rule = "Number1"}

Rule: Number2
(
({Sentence,Sentence contains Physical_Damage }):sentence
)
-->
:sentence. Physical_Damage _Sentence={rule = "Number2"}

... ... ... ...
```

Fig. 3. Snippet of JAPE grammar.

were obtained through document analysis (natural disaster news articles). To populate the entities in the gazetteer list, we utilized the WordNet thesaurus which could provide synonyms or semantically related concepts of the entities. Once the text documents are annotated with respect to these entities, then by using Java Annotation Patterns Engine (JAPE) grammar; as given in Fig. 3, the component sentences are recognized and extracted. We employ the General Architecture for Text Engineering (GATE) tool [31], which is a widely used NLP framework that provides the platform to perform this task. All extracted sentences are then categorized into its corresponding component clusters based on the annotation tags. Note that the sentences that do not belong to any of the component clusters will be discarded at this stage.

We have altogether seven component clusters representing the disaster domain i.e. *description of event*, *human damages*, *physical damages*, *relief efforts*, *persons*, *locations* and *organizations*. It should be noted that the sentences in {*persons*, *locations*, *organizations*} are subsumed by {*description of event*, *human damages*, *physical damages*, *relief efforts*}. However all these sentences will be evaluated with respect to their individual clusters and their scores will only be merged prior to sentence selection. This process will be discussed later in Section 5.2: sentence scoring and selection.

4. Cross-document relation identification

As we mentioned earlier in this paper, in topically related documents, especially news articles, its information contents are closely connected even though the news story comes from different sources. For instance, Fig. 4 depicts the existence of sentence links between different document sources. Here, we will investigate the utility of cross-document or CST relations for identifying highly relevant sentences to be included in the summary. In our work, the sentence relevancy is evaluated with respect to its component cluster. We have considered four types CST relations, namely *Identity*, *Subsumption*, *Description* and *Overlap*; as they cover most of the other relations in the CST model. Descriptions of the four CST relations are provided in Table 1.

Relying on manually annotated text for CST relation identification can consume time and resources. This has motivated us to automatically identify the four aforementioned CST relations to facilitate our multi document summarization task. In our previous work [32], we modeled a case based reasoning (CBR) classifier to identify the CST relationships between sentences and found

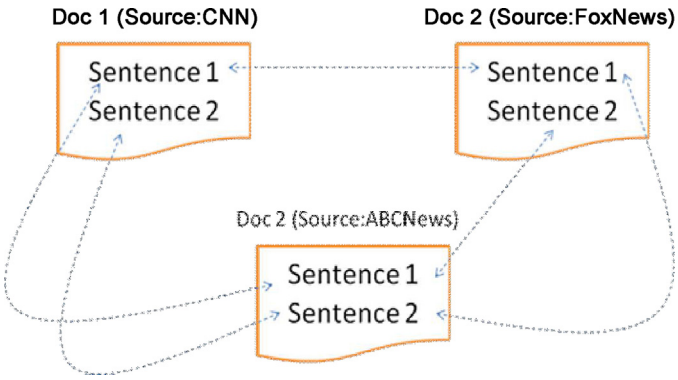

Fig. 4. Sentence links between different document sources.

that the CBR classifier performed well compared to the state of the art. However, in this work we propose a novel approach, named Genetic-CBR to further improve our CBR model because the naïve CBR treated all features equally. The following sub-section describes this approach.

4.1. Generic-CBR approach

"Case-Based Reasoning (CBR) is the usual name given to problem solving methods which make use of specific past experiences. It is a form of problem solving by analogy in which a new problem is solved by recognizing its similarity to a specific known problem, then transferring the solution of the known problem to the new one" [33]. CBR has been applied to solve various real world problems such as course timetabling, solving legal cases and classifying the disease of a patient [34]. We could also regard CBR as a type of supervised learning method as it finds solutions for new problems based on existing solutions.

The general process of CBR consists of four major phases, namely *Retrieve*, *Reuse*, *Revise*, and *Retain* that links to a central repository called the casebase [35]. When a new case (problem) is received, the CBR model will first retrieve the most similar cases from the casebase (where previous solved cases are stored) and the solution from the retrieved cases will be reused for the new case. If no similar cases are found in the casebase, the solution for the new case will be revised and retained into the casebase as a new solved case.

In our work, we consider the task of identifying the CST relation between sentence pairs as a multiclass classification problem, whereby the relation can be classified to one of the following: *Identity*, *Subsumption*, *Description*, *Overlap* or *No Relation*. The inclusion of *No Relation* is necessary as we cannot assume all sentence pairs to be related. Our method incorporates the adaptation of the standard CBR algorithm that is tailored to the classification task. Each case in our casebase represents an example of sentence pair with its known CST relationship type. Specifically, every sentence pair is labeled by its feature vector as shown in Table 2. Next we describe the features that represent each sentence pair:

Table 1
Description of CST relations used in this work.

Relations	Description
Identity	The same text appears in more than one location
Subsumption	S1 contains all information in S2, plus additional information not in S2
Description	S1 describes an entity mentioned in S2
Overlap (partial equivalence)	S1 provides facts X and Y while S2 provides facts X and Z; X, Y, and Z should all be non-trivial.

Table 2
An example of case representation.

Cases	Features					Relation type
	CS	WO	LT	NP	VP	
Case 1	0.23	0.36	0	0.27	0.16	Description
Case 2	0.44	0.34	1	0.55	0.36	Subsumption

Cosine similarity (CS) – cosine similarity is used to measure how similar two sentences (S) are. Here the sentences are represented as word vectors with *tf-idf* as its element (i) value:

$$\cos(S_1, S_2) = \frac{\sum S_{1,i} \cdot S_{2,i}}{\sqrt{\sum (S_{1,i})^2} \cdot \sqrt{\sum (S_{2,i})^2}} \quad (1)$$

Word overlap (WO) – this feature represents the measure based on the number of overlapping words in the two sentences. This measure is not sensitive to the word order in the sentences:

$$\text{overlap}(S_1, S_2) = \frac{\#\text{commonwords}(S_1, S_2)}{\#\text{words}(S_1) + \#\text{words}(S_2)} \quad (2)$$

Length type (LT) – this feature gives the length type of the first sentence when the lengths of two sentences are compared.

$$\begin{aligned} \text{length type}(S_1) \quad &= 1 \quad \text{if} \quad \text{length}(S_1) > \text{length}(S_2), \\ &-1 \quad \text{if} \quad \text{length}(S_1) < \text{length}(S_2), \\ &0 \quad \text{if} \quad \text{length}(S_1) = \text{length}(S_2) \end{aligned} \quad (3)$$

NP similarity (NP) – this feature represents the noun phrase (NP) similarity between two sentences. The similarity between the NPs is calculated according to Jaccard coefficient as defined as in the following equation:

$$\text{NP}(S_1, S_2) = \frac{\text{NP}(S_1) \cap \text{NP}(S_2)}{\text{NP}(S_1) \cup \text{NP}(S_2)} \quad (4)$$

VP similarity (VP) – this feature represents the verb phrase (VP) similarity between two sentences. The similarity between the VPs is calculated according to Jaccard coefficient as defined as in the following equation:

$$\text{VP}(S_1, S_2) = \frac{\text{VP}(S_1) \cap \text{VP}(S_2)}{\text{VP}(S_1) \cup \text{VP}(S_2)} \quad (5)$$

To determine the relationship type for a new case, the model will compare the feature vector of the new case with existing cases in the casebase. In our implementation, we use the cosine similarity measure to compute the similarity between two cases. However, in this work we will assign weights to the features so that the performance of the CBR classification model can be improved; as CBR is an instance based learning method in which its similarity function is very sensitive to the relevance of features used. In order to obtain the weights, we have integrated feature weighting using genetic algorithm; details in Section 4.2. Algorithm 2 below describes the Genetic-CBR implementation while Fig. 5 illustrates the overall process flow.

Algorithm 2. Genetic-CBR Implementation for CST Relation Identification

1. *Input sentence set S*: take the sentences from the component cluster as input, $S = \{S_1, S_2, S_3, \ldots, S_n\}$
2. *Generate sentence pairs*: altogether there are $n^2 - n$ possible sentence pairs in a component cluster of size n.
3. *Features extraction*: extract the features $F = \{CS, WO, LT, NP, VP\}$ from each sentence pair.
4. Retrieve similar cases:
   4.1 Retrieve similar cases from the casebase using the weighted cosine similarity measure: Eq. (6). The weights are obtain using the genetic-based learning algorithm (refer to Algorithm 3).

$$w\cos(X, Y) = \frac{\sum_{k=1}^{5} w_k x_k \times w_k y_k}{\sqrt{\sum_{k=1}^{5} (w_k x_k)^2} \times \sqrt{\sum_{k=1}^{5} (w_k y_k)^2}}, \quad (6)$$

   where $w\cos(X,Y)$ denotes the similarity between two cases and w_k is the weight of the kth feature, $k = \{1,2,\ldots,5\}$.
5. Determine the CST relation of the new case:
   5.1 If the similarity value of the new case is more than the predefined threshold value, the model will reuse the solution i.e. the CST relation.
   5.2 If the similarity value is less than the threshold value, the model will revise the new case solution as "No relation" and retain the revised case into the casebase.
6. Repeat steps 4–5 for each sentence pair in the component cluster.
7. Repeat steps 1–6 for each component cluster.

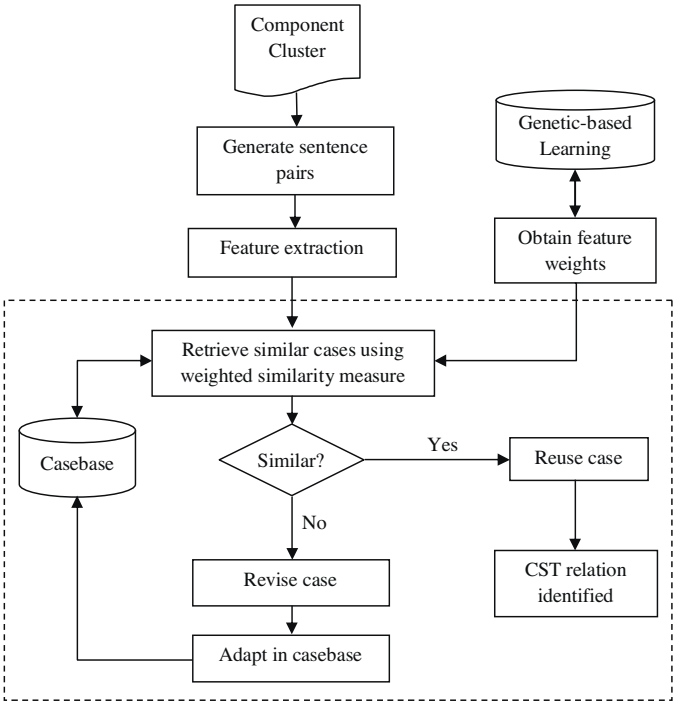

Fig. 5. Genetic-CBR approach for CST relationship identification.

4.2. Feature weighting using genetic-based learning algorithm

As described in Section 4.1, the CBR classifier relies on similarity-based selection (in our case the similarity between feature vectors) to retrieve similar cases from the casebase. Moreover in existing setting, all features are assumed to hold equal importance (weight) by the classifier. These observations, taken together with the fact that similarity functions are generally sensitive to the relevance of features used, suggest that scaling the relevance of features is crucial for the success of the classifier.

In this work, we employ genetic learning algorithm (GA) to optimize the weights of the features used by the CBR classifier. GA is a well known optimization technique used in various fields of research and applications [36–38]. It is based on the evolution theory with the analogy that better solution can be build if we somehow combine the "good" parts of other solutions to generate the new ones. Algorithm 3 describes the genetic learning procedure while Fig. 6 illustrates the process flow.

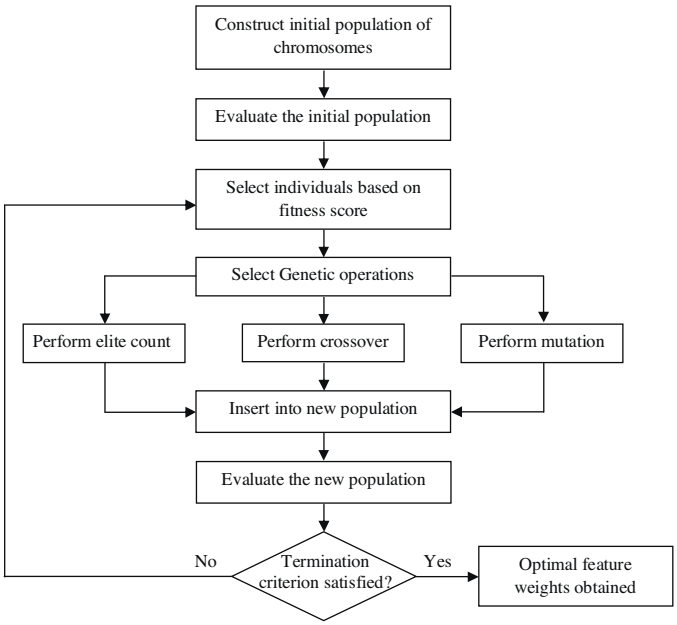

Fig. 6. Genetic algorithm procedure.

Algorithm 3. Genetic-based learning algorithm to find the optimal feature weights

1. Construct initial population:
   1.1 Represent the weight vector for features $F = \{CS, WO, LT, NP, VP\}$ as a chromosome, where the entry of each gene is a real-valued number in the range [0,1].
   1.2 Set the size of chromosome population and initialize each chromosome with random values within the predefined range.
2. Evaluate the population:
   2.1 Apply the chromosome values as weights to the similarity function in the CBR model.
   2.2 Using the training data, determine the CBR model classification accuracy: Eq. (8) and use it to determine the fitness of the chromosome: Eq. (9).
   2.3 Repeat steps 2.1–2.2 for each individual chromosome in the population.
3. *Parents selection*: choose parents who will contribute to the next generation based on Stochastic selection (see details in Section 4.2.3). The number of parents that will be required is:

$$\#\text{Parents} = 2 * \#\text{Crossover child} + \#\text{Mutation child} \quad (7)$$

4. Perform reproduction operations: create new population.
   4.1 Elite count: select the individuals with the best fitness values from the current population.
   4.2 Crossover: use scattered method for the crossover operation (see details in Section 4.2.4).
   4.3 Mutation: use Gaussian mutation (see details in Section 4.2.4)
5. Repeat steps 2–4 until 100 generations.
6. *Obtain feature weights*: the optimal feature weights are selected.

4.2.1. Chromosome and initial population construction

The first step in GA requires the construction of initial population which is composed of chromosomes. Each individual chromosome represents a potential solution to the given problem – in our case, the weights of the features. Since we have five features to be weighted, we construct five dimensional weight vectors by randomly initializing them with values between 0 and 1. These

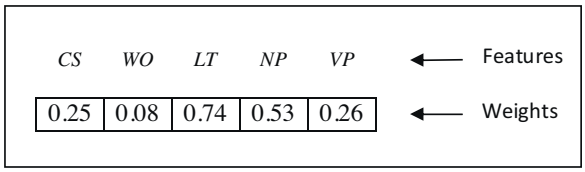

Fig. 7. Structure of chromosome.

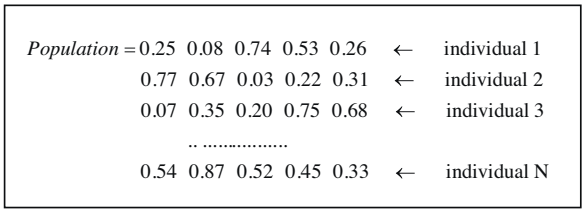

Fig. 8. Chromosome population.

are known as real-valued population. Fig. 7 shows the structure of a chromosome while Fig. 8 shows an example of chromosome population.

4.2.2. Fitness function design

The next thing we need to provide to a GA model is the fitness function, i.e. the function that needs to be optimized. Fitness function is used to reflect the quality of each individual chromosome in the population by computing its fitness value. In this work, we use our CBR model classification accuracy to evaluate the fitness of individuals. The classification accuracy is given by:

$$\text{Classification accuracy} = \frac{c}{t} \quad (8)$$

where c is the number of sentence pairs correctly classified and t is the total number of sentence pairs in the training set. Since GA minimizes its fitness function, we use the classification error rate as the fitness function:

$$\text{Error rate} = 1 - \text{classification accuracy} \quad (9)$$

4.2.3. Selection operation

The selection operation describes how individuals are selected to become parents, to produce the population for the next generation. We use the *Stochastic universal sampling* (SUS) method to select the parents from the current population. SUS uses M equally spaced steps in the range [0,*Sum*], where M is the number of selections required and *Sum* is the sum of the scaled fitness values over all the individuals in the current population. The M parents are then chosen by moving along the above range in steps of *Sum*/M and select the individual whose fitness spanned by each step.

4.2.4. Reproduction operations

The reproduction determines how the GA process creates children at each new generation. We use altogether three reproduction operations i.e. elite count, crossover and mutation. Elite count selects the individuals with the best fitness values in the current population that are guaranteed to survive to the next generation. Crossover operator produces children by mating its parent, i.e. by crossing parts of chromosomes from both parents, while the mutation operation is performed by randomly replacing the gene of the chromosome by another to produce a new genetic structure.

In our implementation, we use scattered method for the crossover operation and Gaussian mutation for the mutation operation. Scattered crossover first creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and

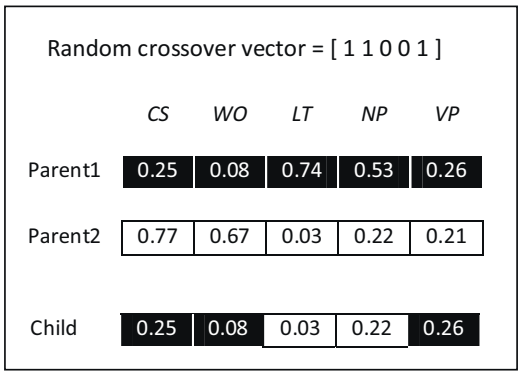

Fig. 9. Reproduction based on scattered crossover.

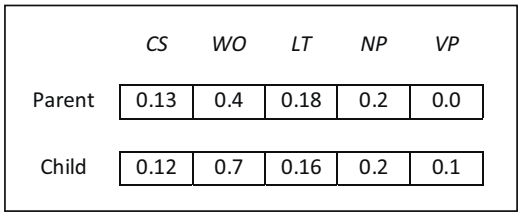

Fig. 10. Gaussian mutation of parent to form a child.

combines the genes to form the child. An example is illustrated in Fig. 9. Gaussian mutation on the other hand, adds a random number, or *mutation*, chosen from a Gaussian distribution, to each entry of the parent chromosome to create a new offspring as shown in Fig. 10. The amount of mutation, which is proportional to the standard deviation of the distribution, decreases at each new generation.

4.2.5. Termination criteria

A stopping or termination criterion determines what causes the algorithm to terminate. We continue to generate new generations and evaluate their fitness until the process reaches the maximum iteration (maximum generation). When the process ends, the individual chromosome with the best fitness value will be selected as optimal feature weights.

5. Fuzzy reasoning for sentence scoring

So far we have discussed the first two of three main phases involved in our proposed summarization model; i.e. the component sentence extraction and cross-document relation (CST relation) identification. The outcomes from these two phases will give us the distribution of CST relations based on its type for each sentence in the component clusters. For instance, Table 3 shows an example of sentence pairs relations for each component sentence from a component cluster (with size *N*) where every column pair represents a sentence pair together with its relation. Likewise, such information can be obtained from all the other component clusters as well. However, not all CST relation types have positive effect toward summary generation [21]. This observation motivates us to propose a fuzzy reasoning mechanism to score the sentences based on the type of CST relations they hold.

Fuzzy reasoning is a natural method for knowledge representation based on degrees of membership. One of the advantages of fuzzy reasoning systems is that they describe fuzzy rules, which fit the description of real-world processes to a greater extent. Another advantage lies in their interpretability; it means that it is possible to explain why a particular value appeared at the output of a fuzzy system. This fits our purpose, since the type of CST relation determines the score of a sentence; for example, a sentence with high *description* relation is considered less important to be included in the summary. Thus, employing fuzzy rules can tolerate the ambiguity and imprecise values for choosing the scores of the sentences.

5.1. Fuzzy reasoning implementation

In this sub-section, the fuzzy reasoning implementation will be explained. The main components of a fuzzy reasoning system are: fuzzy knowledge base; fuzzy rule base; inference engine; fuzzifier; and defuzzifier; as illustrated in Fig. 11.

The fuzzy knowledge base is defined by means of fuzzy concepts, including linguistic variables and linguistic terms. Linguistic variable defines the fuzzy concept, for example, *temperature* whose linguistic terms are words in a natural language (*cold*, *warm* or *hot*). In our knowledge base, we have four input linguistic variables – *Identity*, *Subsumption*, *Description* and *Overlap*. Each of these four variables has three linguistic terms, namely *Low*, *Medium* and *High*. Besides that we also have one output linguistic variable called *Sentence Score* with five linguistic terms, namely *Very Low*, *Low*, *Medium*, *High* and *Very High*.

Another component in the fuzzy reasoning system is called the rule base. Generally, the rule base describes the knowledge base, in the form of fuzzy *if-then* rules. Due to their concise form, fuzzy rules are employed to capture the imprecise way of reasoning that plays an essential role in the human ability to make decisions. In particular, our system uses 81 fuzzy rules which are built by domain experts corresponding to all possible combinations of input terms ($\#terms^{\#variables} = 3^4 = 81$). Based on these rules, the score for each sentence is determined according to its input values. For example, *if identity is high and subsumption is high and description is low and overlap is high then sentence score is very high.* Table 4 lists part of the constructed fuzzy rules in our rule base.

The core of a fuzzy reasoning system is its inference engine; which merges the facts obtained from the fuzzy knowledge base with a series of production rules from the rule base. In our implementation, we use Mamdani's inference method [39] to perform the fuzzy reasoning process. All inputs to the inference engine will be first transformed into fuzzy values (membership degrees) using a membership function (fuzzifier). There exist a variety of membership function that comes with different shapes such as Triangular, Trapezoidal, Bell, Gaussian, Polynomial and Sigmoidal.

Table 3
Example of sentence pairs relations for a component cluster with size N.

Sentence 1	Relation	Sentence 2	Relation		Sentence N	Relation
S1 → S2	Identity	S2 → S1	Identity	...	SN → S1	Overlap
S1 → S3	No relation	S2 → S3	Description	...	SN → S2	No relation
S1 → S4	Overlap	S2 → S4	No relation	...	SN → S3	Subsumption
...	...	...	...	...	...	...
S1 → SN	Overlap	S2 → SN	No relation	...	SN → SN-1	Description

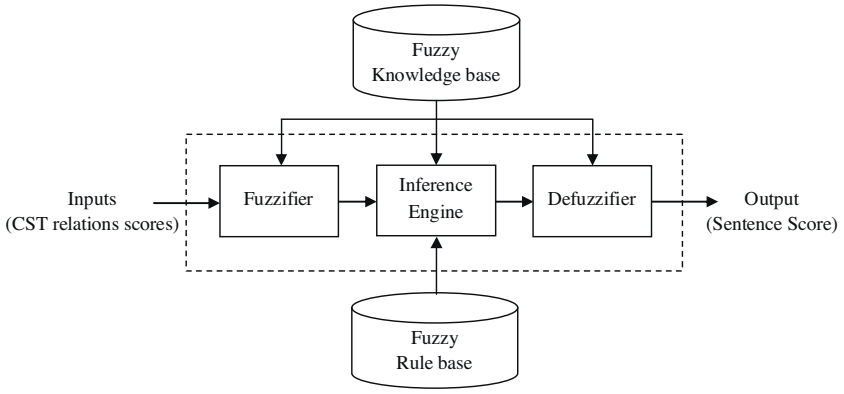

Fig. 11. Fuzzy reasoning system.

Table 4
Part of the constructed fuzzy rules.

Rule no.	Input fuzzy variables				Output fuzzy variable Sentence Score
	Identity	Subsumption	Description	Overlap	
R1	Low	Low	Low	Low	Very Low
R2	Low	Low	Low	Medium	Medium
R3	Low	Low	Low	High	High
R4	Low	Low	Medium	Low	Low
R5	Low	Low	Medium	Medium	Medium
R6	Low	Low	Medium	High	High
R7	Low	Low	High	Low	Very Low
R8	Low	Low	High	Medium	Low
R9	Low	Low	High	High	High
R10	Low	Medium	Low	Low	Low
...	...	...	...	...	...
R79	High	High	High	Low	Low
R80	High	High	High	Medium	Medium
R81	High	High	High	High	Very High

For this work, we use the Gaussian membership function because of its simplicity and robustness [40]. Figs. 12 and 13 show the Gaussian membership functions for input fuzzy variable *Description* and output fuzzy variable *Score*, respectively. We determined the parameters of all membership functions as defined by the expert (linguist). After performing the fuzzy inference mechanism, defuzzification is carried out to convert fuzzy numbers into representative crisp values. These steps are described in Algorithm 4.

Algorithm 4. Fuzzy Reasoning Implementation

1. *Input CST relation scores*:

1.1 Give each CST relation {*Identity*, *Subsumption*, *Description*, *Overlap*} its score based on the distribution of its relation for that sentence.

$$\text{Score}\ (S_{i,n}) = \frac{\sum_{j=1}^{|C|} R_{i,j}^{n}}{\sum_{n=1}^{4}\sum_{j=1}^{|C|} R_{i,j}^{n}} \tag{10}$$

$$\text{where } R_{i,j}^{n} = \begin{cases} 1, & \text{if relationship } n \text{ exist}, \quad i \neq j \\ 0, & \text{otherwise} \end{cases}$$

Score ($S_{i,n}$) is the score of relation n for sentence i. C is the set of sentences in the component cluster.

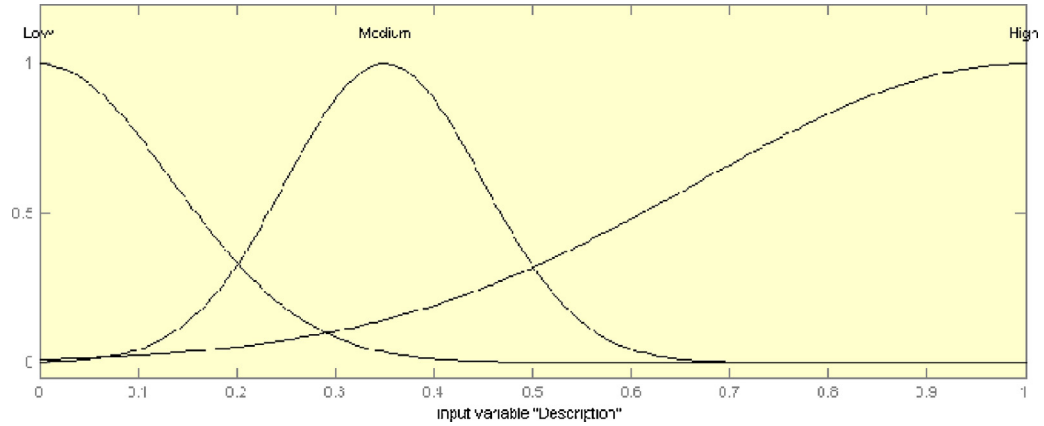

Fig. 12. The Gaussian membership functions for the input variable *Description*.

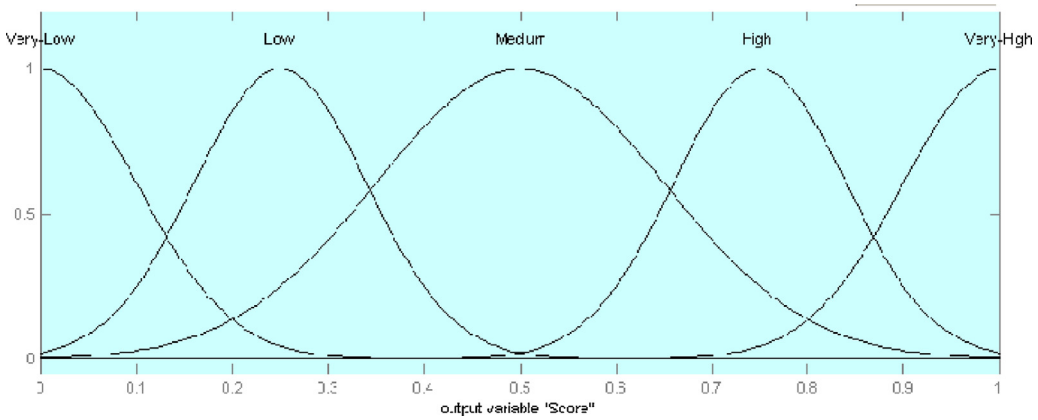

Fig. 13. The Gaussian membership functions for the output variable *Score*.

1.2 Use the obtained scores as input to the fuzzy reasoning model.

2. *Fuzzification*: use the Gaussian membership function: Eq. (11) to compute the membership degrees for the crisp score, x of each CST relation.

$$g(x) = e^{-((x-c)^2/2\sigma^2)} \tag{11}$$

where parameters c and σ determine the center and the shape of the curve, respectively. The values $c = 0$ and $\sigma = 1$ define the standard Gaussian membership function.

3. *Inference*:
   3.1 The AND operator is used to combine the degree of match between each fuzzy rule's conditions.
   3.2 Use the Gaussian membership function: Eq. (11) to represent the output of each rule.
   3.3 Aggregate the output of each rule into a single fuzzy set.
4. *Defuzzification*: resolve a single output value (representing sentence score) from the set fuzzy set using the centroid calculation: Eq. (12), which returns the center of area under the curve.

$$\text{Score}_{\text{fuzzy}} = \frac{\int \mu_i(x)x\,dx}{\int \mu_i(x)dx} \tag{12}$$

where $\mu_i(x)$ is the aggregated membership function and x is the output variable.

5. Repeat steps 1–4 for each sentence in the component cluster.
6. Repeat steps 1–5 for each component cluster.

5.2. Sentence scoring and selection

The final process prior to summary generation involves sentence scoring and selection. We compute the final score of each sentence in all component clusters by aggregating two scores; (1) the sentence score obtained based on the total number of CST relation and; (2) the score produced by the fuzzy reasoning system, which is based on the CST relation type. We use the algebraic product method to aggregate both scores. This score is given as follows:

$$\text{Score}\ \ (S_i) = \frac{\sum_{n=1}^{4}\sum_{j=1}^{|C|} R_{i,j}^{n}}{|S|-1} \times \text{Score}_{\text{fuzzy}} \tag{13}$$

The notations for the above equation are similar to that of Eq. (10). Through this aggregation, a sentence with high number of CST relations could still be penalized if its corresponding result obtained from the fuzzy reasoning system indicates low score. This scoring method balances the size of relation and the type of relation a sentence holds. After computing the aggregated score for all sentences in the component clusters, we further combine the scores for overlapping sentences between $A = \{persons, locations, organizations\}$ and $B = \{description\ of\ event, human\ damages, physical\ damages, relief\ efforts\}$. As mentioned earlier in Section 3.2, the sentences in set A are subsumed by set B. Hence, we use the probabilistic sum method to aggregate their scores:

$$\text{Score total}(S_i) = Com_{A,i} + Com_{B,i} - Com_{A,i}Com_{B,i} \tag{14}$$

where Com_A is the score of sentences from component set A and Com_B is the score of sentences from component set B. We use the document and sentence number as reference index to perform sentence matching between the two sets. Eq. (14) increases the score of sentence i, if the contextual information reflecting the person, location or organization in that sentence is important. Once the scores between these two sets of component clusters have been merged, we re-rank the sentences in set B using their updated scores. Finally, high ranking sentences are selected from each component cluster (set B) until the desired summary length is met.

6. Evaluation and results

The purpose of the evaluation is twofold: first, to assess the performance of the Genetic-CBR model for the task of cross-document relation identification; and second, to evaluate the overall performance of the summarization model which is based on news components using fuzzy cross-document relations.

6.1. CST relation identification experiments

To conduct the experiment for CST relation identification, we used the dataset obtained from CSTBank [41] – a corpus consisting clusters of English news articles annotated with CST relationships. We collected 582 sentence pairs having the relation types *Identity*, *Subsumption*, *Description* and *Overlap*. We also manually selected 100 pairs of sentences that hold no CST relations. The features (as described in Section 4.1) are extracted from each sentence pair. These features will then form the instances for the training and test set where each instance is represented as feature vector paired with its corresponding CST relationship type.

We first run Algorithm 3 to find the optimal weights for the features. The detail parameter settings for the algorithm are as follows: the initial population consists of 20 chromosomes which were randomly initialized with real values between 0 and 1. To evaluate the fitness function i.e. the CBR classification error rate, each fitness value were obtained using the average classification accuracy by running 10 hold-out cross validation on the training set. Selected chromosomes were then reproduced, resulting 2 elite child, 11 crossover child and 7 mutation child in each generation. We run 100 generations (to allow the fitness to converge) before we terminate the process.

Table 5
Precision, recall, and *f*-measure of Genetic-CBR classification.

CST type	Precision	Recall	*f*-Measure
No relation	0.96	0.8	0.872727
Identity	1	0.966667	0.983050
Subsumption	0.837209	0.72	0.774193
Description	0.760869	0.921053	0.833333
Overlap	0.730158	0.793103	0.760330

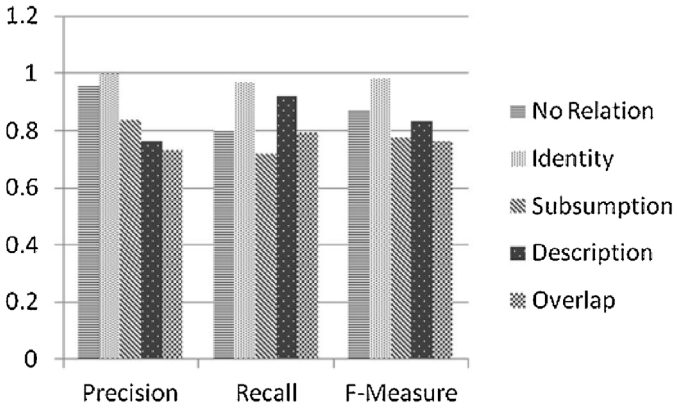

Fig. 14. Performance of Genetic-CBR classification.

With Algorithm 3, optimal weights are outputted. The optimal weights are 0.18374, 0.94211, 0.81638, 0.61879 and 0.00631, representing the weights for cosine similarity, length type, word overlap, noun phrase similarity and verb phrase similarity, respectively. These values reflect the significance of the features. For example length type is highly significant and verb phrase is less significant. We then use these results for the weighted cosine similarity function in our Genetic-CBR classifier. We employ the evaluation measures commonly used in classification tasks – Precision, Recall and *f*-measure. Table 5 and Fig. 14 show the precision, recall, and *f*-measure of Genetic-CBR classification.

To compare our method with other methods, we also tested the performance of CBR (without feature weighting), neural network (NN) and support vector machine (SVM). NN and SVM are two popular machine learning techniques used for classification tasks. Furthermore SVM has been used in the literature concerning multi-class relationship identification and was found to outperform other classifiers such as decision tree and Naïve Bayes [26]. In this experiment, the parameters of NN and SVM were tuned to give optimal results. To evaluate to the SVM classifier, we trained the data using the LibSVM tool developed by Chang and Lin [42]. LibSVM is an integrated software which is extensively used for solving multi-class classification problems. For the kernel selection, we chose the RBF kernel function as it gives better accuracy and as stated in [43], it has several advantages over the other kernel functions. The SVM model best parameters (c, cost of misclassification penalty and g, gamma of RBF kernel function) were chosen after applying 5-fold cross validation. Once the training is completed, the resulting classifier model is then tested with the test data to measure its performance.

To evaluate the performance of NN, we trained the data using the Neural Network tool on MATLAB. We use a multi-layer feed-forward network which has been proven to be the universal function approximator [44], with the most popular back-propagation learning algorithm. The number of hidden nodes H_i is initially set to 1. The accuracy of the network is then recorded for H_i after training it. Then H_i is incremented and the process continues. The process ends when the result of H_i is better than H_{i+1} and H_{i+2}. After determining the best H, we fixed it as the number of hidden node in the network hidden layer. The resulting network model is then tested with the test data to measure its performance. The classification accuracies of these methods are shown in Fig. 15. It can be observed that the accuracies of NN and SVM are lower than CBR while Genetic-CBR obtained the best accuracy i.e. 82.52%.

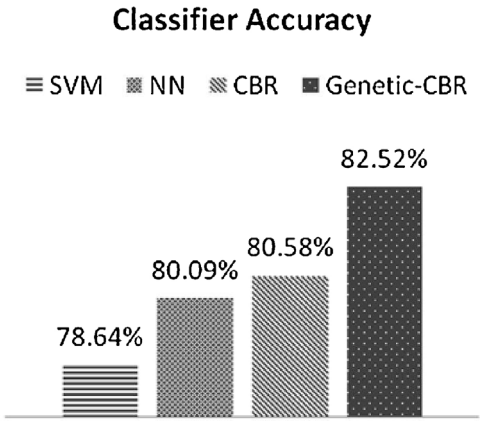

Fig. 15. Accuracy comparison between SVM, NN, CBR and Genetic-CBR.

6.2. Summarization experiments

This section describes the evaluation of our proposed summarization model. We evaluate the proposed model using the DUC 2002 document sets (D061j, D062j, D073b, D077b, D079a, D083a, D085d, D089d, D091c, D092c, D097e, D103g, D109h and D115i) comprising 117 documents corresponding to natural disaster news stories. The Document Understanding Conference (DUC) is a standard corpus used in text summarization studies which contains documents and with their human model summaries. The evaluation results (recall, precision and *f*-measure) are obtained using ROUGE: Recall-Oriented Understudy for Gisting Evaluation [45]. ROUGE measures the quality of a system generated summary by comparing it to a human model summary (H2). There are many variances in ROUGE evaluation measure; however it was found that ROUGE-1, ROUGE-2, ROUGE-S and ROUGE-SU worked well in multi document summarization tasks [45]. Next we define the performance measures of ROUGE-N (in this case, $N = 1, 2$).

ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries, and is computed as follows:

$$\text{ROUGE} - N = \frac{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{gram}_n \in S} \text{count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{Reference summaries}\}} \sum_{\text{gram}_n \in S} \text{count}(\text{gram}_n)} \quad (15)$$

where n is the length of the n-gram, gram_n and $\text{count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. The description of the other measures (ROUSE-S and ROUGE-SU) can be found in Ref. [45].

To compare the performance of our model, we set up six comparison models besides the human with human benchmark (H1–H2) (H1 against H2). As stated in the literature, there are two mainstream approaches toward multi document summarization tasks, i.e. using cluster based method and graph based method. We built the comparison models based on these two methods. The cluster based method employs the widely used k-means clustering algorithm to generate clusters; from which the summary sentences will be selected. For the graph based method, we rank the sentences using the popular PageRank algorithm. Table 6 below describes all the methods that are evaluated in this experiment.

Tables 7–10 show the comparison between the proposed model (F1) and the other seven methods (H1, C1, C2, C3, G1, G2, G3) based

Table 6
Description of summarization methods evaluated in this work.

Summarization method	Abbr.	Description
FUZZY CST with COM	F1	The proposed method; based on news component using fuzzy CST based ranking.
H1-H2	H1	Human with human benchmark.
CST with COM	C1	Method based on news components using CST based ranking.
CST without COM	C2	Method based on CST based ranking without integrating news components.
CLUSTER with CST	C3	Method based on clustering using CST based ranking.
GRAPH with COM	G1	Method based on news components using graph based ranking.
GRAPH without COM	G2	Method based on graph based ranking without integrating news components.
CLUSTER with GRAPH	G3	Method based on clustering using graph based ranking.

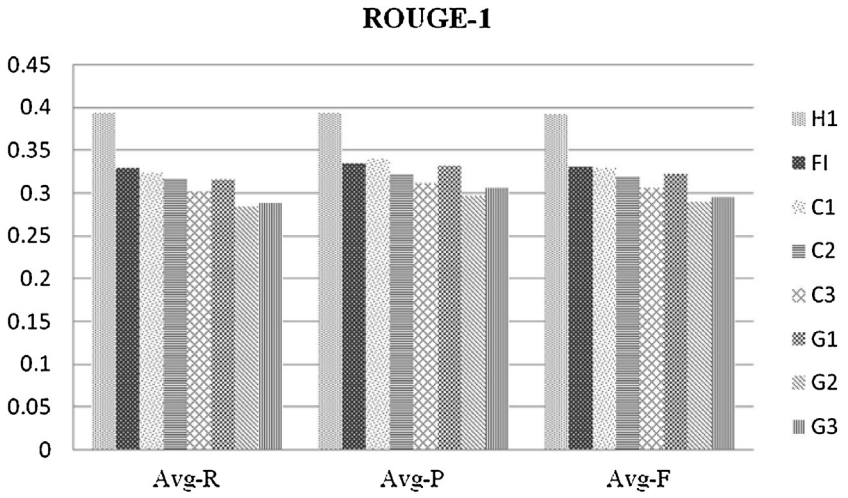

Fig. 16. Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-1.

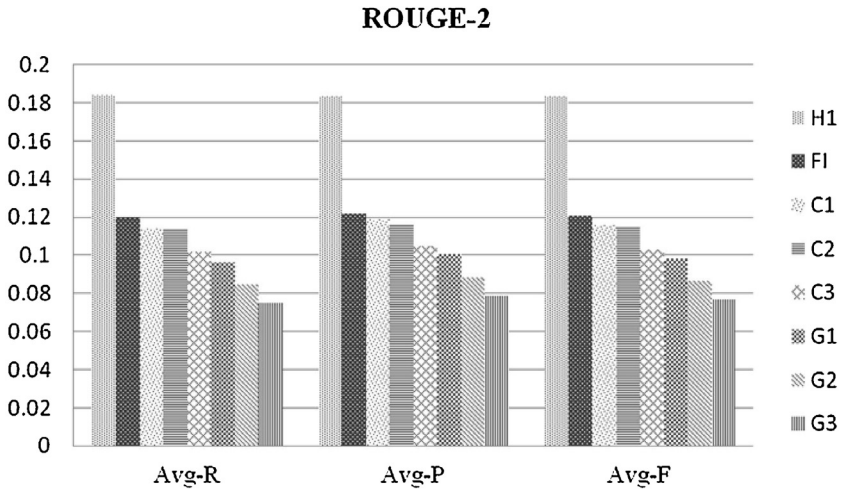

Fig. 17. Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-2.

Table 7
Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-1.

Method	AVG-R	AVG-P	AVG-F
H1	0.39419	0.39402	0.39283
F1	**0.33206**	**0.34101**	**0.33568**
C1	0.32401	0.34038	0.33041
C2	0.31683	0.32321	0.31903
C3	0.30275	0.31187	0.30633
G1	0.31619	0.33220	0.32317
G2	0.28493	0.29736	0.28995
G3	0.28885	0.30672	0.29627

The bold values are obtained with statistically significant improvements at 95% confidence.

Table 8
Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-2.

Method	AVG-R	AVG-P	AVG-F
H1	0.18393	0.18380	0.18332
F1	**0.12806**	**0.12986**	**0.12870**
C1	0.11426	0.11918	0.11616
C2	0.11380	0.11621	0.11470
C3	0.10225	0.10508	0.10337
G1	0.09636	0.10100	0.09841
G2	0.08528	0.08876	0.08667
G3	0.07537	0.07917	0.07690

The bold values are obtained with statistically significant improvements at 95% confidence.

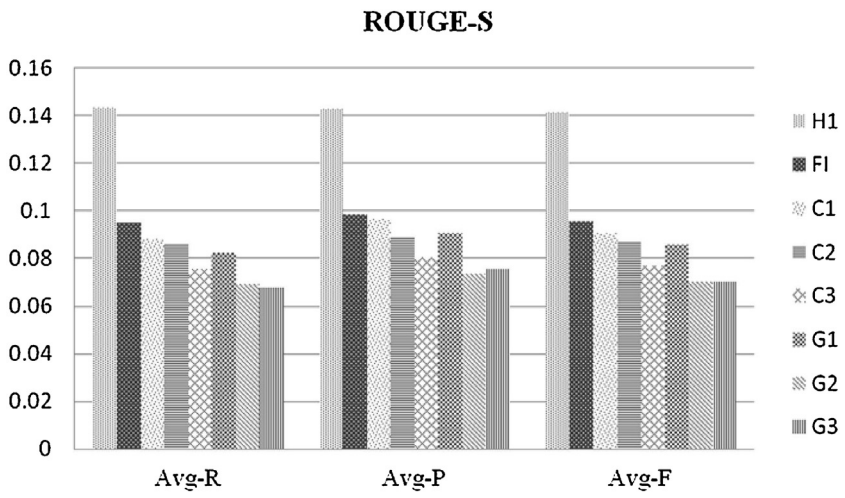

Fig. 18. Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-S.

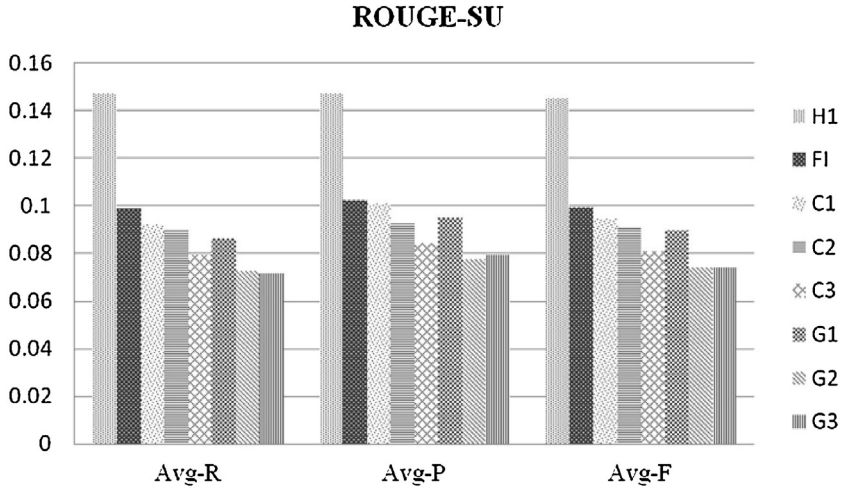

Fig. 19. Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-SU.

on the average recall, precision and *f*-measure using ROUGE-1, ROUGE-2, ROUGE-S and ROUGE-SU respectively. Figs. 16–19 visualize these results. From all these results, it could be observed that the proposed model (F1) performed better than the comparison models and came second to the human benchmark (H1).

6.3. Discussion

The experimental results lead to two main observations: first, the performance of Genetic-CBR approach for cross-document relation identification; and second, the overall performance of the proposed multi document summarization model which is based on news components using fuzzy cross-document relation. To address the first observation, we set forth the effectiveness of the Genetic-CBR model from two aspects: its underlying classification method (i.e. based on case based reasoning) and its final optimized model.

The ability of CBR to surpass the popular SVM and NN for CST relationship classification could be closely related to the nature of its learning method; i.e. lazy learning. As opposed to eager learning methods (such as SVM and NN) which need to generalize

Table 9
Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-S.

Method	AVG-R	AVG-P	AVG-F
H1	0.14330	0.14312	0.14149
F1	**0.09571**	**0.09992**	**0.09695**
C1	0.08842	0.09693	0.09087
C2	0.08616	0.08945	0.08683
C3	0.07601	0.08060	0.07739
G1	0.08268	0.09111	0.08593
G2	0.06924	0.07384	0.07049
G3	0.06815	0.07581	0.07065

The bold values are obtained with statistically significant improvements at 95% confidence.

Table 10
Summarization results comparison based on average recall, precision and *f*-measure using ROUGE-SU.

Method	AVG-R	AVG-P	AVG-F
H1	0.14744	0.14725	0.14560
F1	**0.09965**	**0.10404**	**0.10095**
C1	0.09233	0.10118	0.09490
C2	0.09001	0.09343	0.09070
C3	0.07979	0.08456	0.08123
G1	0.08654	0.09529	0.08992
G2	0.07284	0.07774	0.07419
G3	0.07183	0.07990	0.07447

The bold values are obtained with statistically significant improvements at 95% confidence.

The earthquake that devastated Iran early Thursday took tens of thousands of lives because people were asleep in fragile homes built on flood plains, an earthquake expert said.
Needham said most structures in that area are built of a ceramic-type brick or adobe that collapses easily.
The shock wave traveled through the mountainous section of coastal Iran where most of the buildings are built on a flood plain of loosely deposited soil that shifts in an earthquake and allows structures to collapse, he said.
The 7.7-magnitude quake was the largest ever recorded in that area, where two major plates of the earth's crust meet, Needhams said.
The estimated 50,000 dead in the Iran earthquake make it the world's fourth deadliest quake in the past half-century.
Iranian officials have said they would welcome aid from all countries except Israel and South Africa.
Countries worldwide, both friendly and hostile to Tehran's Islamic government, sent supplies, medical personnel and condolences to Iran, where about 40,000 people have died in an earthquake.
Writer Salman Rushdie, who has been sentenced to die by Iranian zealots, has donated $8,600 to victims of last week's devastating earthquake in Iran.

Fig. 20. Example of human produced summary.

The earthquake that devastated Iran early Thursday took tens of thousands of lives because people were asleep in fragile homes built on flood plains, an earthquake expert said.
Iran said today that it would welcome relief offered by its bitter enemy, the United States, to help victims of the earthquake that has killed as many as 35,000 people, the State Department said in Washington .
Trucks carried aid into earthquake-stricken Iran on Saturday from Soviet Azerbaijan, which observed a day of mourning for victims, the official Tass news agency said.
The shock wave traveled through the mountainous section of coastal Iran where most of the buildings are built on a flood plain of loosely deposited soil that shifts in an earthquake and allows structures to collapse, he said.
Twelve earthquakes greater than magnitude 7 have occurred in Iran during the last 30 years, Needhams said.
In Paris, the French government, which already has sent a 200-member civil defense team including rescuers, medical personnel and search dogs said Saturday it would send an entire mobile hospital unit if Iran wanted it.
The State Department said Friday that Iran was willing to accept earthquake relief from the American Red Cross and other U.S. humanitarian organizations.

Fig. 21. Example of summary generated by the proposed model (F1).

the training data to classify new cases, lazy learning is a learning method which performs classification based on the similarity of a problem with already known problems. Concerning our study, since texts data have high variability, the key advantage of lazy learning is that instead of estimating the target function once for the entire instance space, this method can estimate it locally for each new instance to be classified. On the other hand, the poor performance of SVM compared to the other methods is plausibly due to number of features used, as SVM normally performs well with high-dimensional datasets. This may explain why SVM could not well differentiate between the different classes of CST relations.

The experimental results further showed that better results could be obtained using optimized CBR model: i.e. the Genetic-CBR model. As far as CBR classification method is concerned, the success of the method depends on the retrieval of cases from its case-base; which relies on similarity-based selection. Moreover, CBR is an instance based learning method, in which its similarity function is sensitive to the relevance of features used. This is supported with the fact that the results obtained through genetic algorithm shows that relevance of the features varies. Consequently, the optimization of feature weights has improved its similarity-based selection and the CBR model altogether.

Next, to address the second observation, i.e. the overall performance of our proposed multi document summarization model, we infer the findings of the summarization experiment results. The findings demonstrate that the proposed model (F1) achieved highest score among all comparison models (H1 is excluded for this comparison as it is a human benchmark and was expected to give best results). Two complementary factors have contributed to the success of the proposed model; the inclusion of news components and the integration of fuzzy reasoning over the CST relations.

We believe that taking into account the generic components of a news story; such as *who*, *what*, *when*, *where* and *how* could provide contextual information coverage that would be ideal for news summary creation. Furthermore, such components are very close to how human perceive news contents. The fundamental idea behind the integration of fuzzy reasoning on CST based ranking is to treat the importance of each CST relation based on its relationship type. This is because not all CST relation types have positive effect toward summary generation. For example, a sentence with high *description* relation is considered less important compared to a sentence with high *overlap* relation. See Table 11. However, if we look at the underlying concept of the graph based approach, the 'relation' between sentences is determined based only on its measured similarity value and not based on its relationship type. Such approach assumes all sentences with high similarity to be important sentence regardless of the type of relationship they hold. In this study, we have approached this issue by first identifying the relationship types between the sentence pairs. This allows us to then incorporate the proposed fuzzy reasoning model over the identified

Table 11
Examples of CST relationship between actual sentences (S1 and S2).

Relationship	Sentence (S1)	Sentence (S2)
Overlap	A National Weather Service report said the hurricane was moving west at 17 mph with maximum sustained winds of 115 mph.	Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roofs off homes and buildings, uprooted trees and downed power lines.
Description	A quake of magnitude 7 is considered a major earthquake, capable of widespread, heavy damage in populated areas.	Northern Iran was stuck Thursday by an earthquake measuring between 7.3 and 7.7 on the Richter Scale.

relations. Figs. 20 and 21 shows an example of original human produced summary extract and the corresponding summary generated by our proposed model (F1).

The experimental results also lead to several other important observations with respect to the comparison models (C1, C2, C3, G1, G2 and G3). First; the methods which integrate news components performed better compared to the similar methods that do not integrate news components (C1 vs. C2 and G1 vs. G2). Second; the methods using CST based ranking performed better than the similar methods that use graph based ranking (C1 vs. G1, C2 vs. G2 and C3 vs. G3). Third; the methods that use news components performed better than the similar methods that use clustering approach (C1 vs. C3 and G1 vs. G3). These observations altogether support the idea that the generic news components integrated in news articles and the knowledge obtained from the identified cross-document relations helps to improve the quality of multi document summary.

7. Conclusion

In this paper, we have introduced a multi document summarization model by taking into account the generic components of news story. The study further investigates the utility of cross-document relations (CST relations) to identify highly relevant sentences to be included in the summary. In literature, CST related summarization studies were all based on manually annotated relations by human experts. In this work we have filled this gap by automatically identifying the CST relations between sentences from un-annotated text documents. We achieve this by building a classifier named Genetic-CBR which integrates genetic learning algorithm to the case base reasoning model. The proposed classifier obtained good classification results (with average 85.76% precision, 84.02% recall and 84.47% *f*-measure); making it promising to be integrated into our summarization model. Following that, we develop a new sentence scoring model based on fuzzy reasoning over the CST relations identified by our classifier.

The overall performance of our proposed model was evaluated using the dataset obtained from DUC 2002 whereby its performance was assessed using four ROUGE measures i.e. ROUGE-1, ROUGE-2, ROUGE-S and ROUGE-SU. We also made comparisons with the mainstream methods: cluster based method and graph based method. The experimental findings showed that the proposed model gave better results and supports our hypothesis i.e. providing contextual information coverage using generic components of news would be ideal for news summary creation, as it is close to the way how humans perceive news contents. The integration of fuzzy reasoning over the CST relations also adds further improvement to the results. Although we focus on natural disaster news stories, the concepts and techniques are applicable to other domains as well.

In the context of generating domain knowledge, our next step is to explore how natural language processing techniques can be employed to connect semantic concepts with news components. The semantic structure of sentences representing different components can be mapped to the events in a news story. Thus, providing semantic annotations relevant to the news components could possibly treat a wide range of news topics of interest. For future work, we would also like investigate how other cross-document relations from the CST model can be identified from un-annotated text documents. By being able to well identify the other relations; we can study the utility of those relations to generate better summaries by treating issues related to multi document such as contradictions and historical information.

Acknowledgements

This research is supported by the Ministry of Higher Education (MOHE), Universiti Teknikal Malaysia Melaka (UTeM) and Soft Computing Research Group (SCRG) of Universiti Teknologi Malaysia (UTM).

References

[1] H.P. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (2) (1958) 159–165.
[2] V. Gupta, G.S. Lehal, A survey of text summarization extractive techniques, J. Emerg. Technol. Web Intell. 2 (2010) 258–268.
[3] A. Nenkova, K. McKeown, Automatic Summarization, vol. 5, Foundations and Trends in Information Retrieval, 2011, pp. 103–233.
[4] H. Saggion, T. Poibeau, Automatic Text Summarization: Past, Present and Future, Multi-source, Multilingual Information Extraction and Summarization, Springer-Verlag, Berlin, Heidelberg/New York, 2013, pp. 3–21.
[5] D. Das, A.F.T. Martins, A Survey on Automatic Text Summarization, Literature Survey for the Language and Statistics II Course at Carnegie Mellon University, Pittsburgh, United States, 2007.
[6] K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J.L. Klavans, A. Nenkova, et al., Tracking and summarizing news on a daily basis with Columbia's Newsblaster, in: Proc. Second Int. Conf. Hum. Lang. Technol. Res., 2002.
[7] M. Haque, S. Pervin, Z. Begum, Literature review of automatic multiple documents text summarization, Int. J. Innov. Appl. Stud. 3 (1) (2013) 121–129.
[8] D.R. Radev, A common theory of information fusion from multiple text sources step one: cross-document structure, in: Proc. SIGDIAL, vol. 10, 2000, pp. 74–83.
[9] A. Nenkova, K. McKeown, A Survey of Text Summarization Techniques, Mining Text Data, Springer, US, 2012, pp. 43–76.
[10] D.R. Radev, H. Jing, M. Sty, D. Tam, Centroid-based summarization of multiple documents, in: Proc. Inf. Process. Manage., 2004, pp. 919–938.
[11] X. Xu, A new sub-topic clustering method based on semi-supervised learning, J. Comput. 7 (10) (2012) 2471–2478.
[12] R.M. Aliguliyev, Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization, in: Proc. Comput. Intell., 2010, pp. 420–448.
[13] Y. Xia, Y. Zhang, J. Yao, Co-clustering sentences and terms for multi-document summarization, in: Proc. CICLing, 2, 2011, pp. 339–352.
[14] N. Yu, D. Ji, L. Yang, Z. Niu, T. He, Multi-document summarization using a clustering-based hybrid strategy, in: Proc. AIRS, 2006, pp. 608–614.
[15] G. Erkan, D.R. Radev, LexRank., Graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. (JAIR) 22 (1) (2004) 457–479.
[16] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Seventh International World-Wide Web Conference (WWW), Brisbane, Australia, 1998.
[17] S. Hariharan, T. Ramkumar, R. Srinivasan, Enhanced graph based approach for multi document summarization, Int. Arab J. Inform. Technol. (IAJIT) 10 (4) (2013).
[18] S. Hariharan, R. Srinivasan, Studies on graph based approaches for single and multi document summarizations, Int. J. Adv. Comput. Theory Eng. 1 (5) (2009) 519–526.
[19] X. Wan, An exploration of document impact on graph-based multi-document summarization, in: Proc. EMNLP, 2008, pp. 755–762.
[20] F. Wei, W. Li, Q. Lu, Y. He, A document-sensitive graph model for multi-document summarization, in: Proc. Knowl. Inf. Syst., 2010, pp. 245–259.
[21] Z. Zhang, S. Blair-Goldensohn, D.R. Radev, Towards CST-enhanced summarization, in: Proc. AAAI/IAAI, 2002, pp. 439–446.
[22] M.L.C. Jorge, T.A.S. Pardo, Experiments with CST-based Multidocument Summarization, Workshop on Graph-based Methods for Natural Language Processing, ACL, Uppsala, Sweden, 2010, pp. 74–82.

[23] Z. Zhang, J. Otterbacher, D.R. Radev, Learning cross-document structural relationships using boosting, in: Proc. Twelfth Int. Conf. Inform. Knowl. Manage., 2003, pp. 124–130.
[24] Z. Zhang, D.R. Radev, Combining labeled and unlabeled data for learning cross-document structural relationships, in: Proc. IJCNLP, 2004, pp. 32–41.
[25] N.A.H.B. Zahri, F. Fukumoto, Multi-document summarization using link analysis based on rhetorical relations between sentences, in: Proc. CICLing (2), 2011, pp. 328–338.
[26] E.G. Maziero, T.A.S. Pardo, Automatic Identification of Multi-document Relations (PROPOR 2012 Ph.D. and M.Sc./M.A. dissertation contest), 2012, pp. 1–8.
[27] Y. Miyabe, H. Takamura, M. Okumura, Identifying cross-document relations between sentences, in: Proc. 3rd Int. Joint Conf. Nat. Lang. Process., 2008, pp. 141–148.
[28] J.M. Neal, S.S. Brown, Newswriting and Reporting, Surjeet Publications, Delhi, 1982.
[29] M. Moens, Information Extraction: Algorithms and Prospects in a Retrieval Context, The Information Retrieval Series, Springer-Verlag, Secaucus, NJ, 2003.
[30] D.C. Wimalasuriya, D. Dou, Ontology-based information extraction: an introduction and a survey of current approaches, J. Inform. Sci. 36 (3) (2010) 306–323.
[31] H. Cunningham, K. Bontcheva, V. Tablan, D. Maynard, GATE: a framework and graphical development environment for robust NLP tools and applications, in: Proc. 40th Anniv. Meet. Assoc. Comp. Linguist. (ACL), Philadelphia, 2002.
[32] Y.J. Kumar, N. Salim, B. Raza, Cross-document structural relationship identification using supervised machine learning, Appl. Soft Comput. 12 (10) (2012) 3124–3131.
[33] R. Bareis, Exemplar-based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning, Academic Press, Boston, United States, 1989.
[34] C. Fan, P. Chang, J. Lin, J.C. Hsieh, A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification, Appl. Soft Comput. 11 (1) (2011) 632–644.
[35] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations and system approaches, AI Commun. 7 (1) (1994) 39–59.
[36] W. Paszkowicz, Genetic algorithms, a nature-inspired tool: survey of applications in materials science and related fields, in: Mat. Man. Proc., vol. 24, 2009, pp. 174–197.
[37] M.T. Scott, An introduction to genetic algorithms, J. Comput. Sci. Coll. (ACM) 20 (2004) 115–123.
[38] T. Anita, D. Rucha, Article: genetic algorithm – survey paper, in: IJCA Proc. NCRTC, Found. Comput. Sci., vol. 5, New York, 2012, pp. 25–29.
[39] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man. Mach. Stud. 7 (1975) 1–15.
[40] K. Thangavel, R. Roselin, Fuzzy – rough feature selection with Π-membership function for mammogram classification, Int. J. Comput. Sci. Issues (IJCSI) 9 (4) (2012) 361.
[41] CSTBank PhaseI, http://tangra.si.umich.edu/clair/CSTBank/
[42] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[43] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2003.
[44] K. Kaikhah, Text Summarization Using Neural Networks, Faculty Publications-Computer Science, Texas State University, San Marcos, Texas, United States, 2004.
[45] C.Y. Lin, ROUGE. A package for automatic evaluation of summaries, in: Proc. Workshop Text Summarization ACL, Spain, 2004.