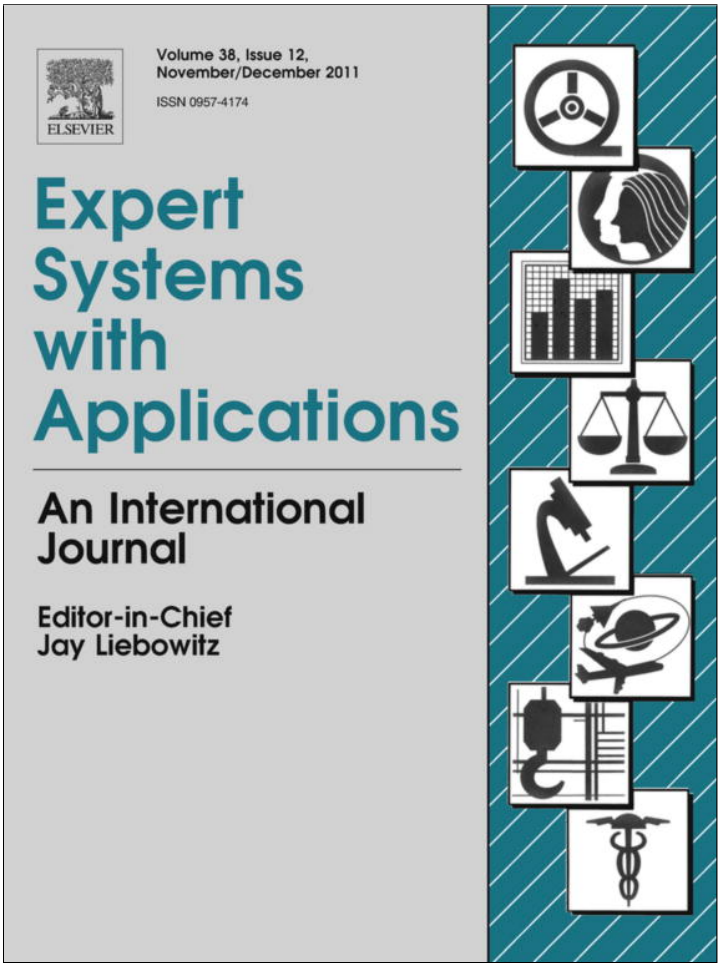
Volume 38, Issue 12,
November/December 2011
ISSN 0957-4174
ELSEVIER
Expert Systems with Applications
An International Journal
Editor-in-Chief
Jay Liebowitz

Expert Systems with Applications 38 (2011) 14514–14522

Contents lists available at ScienceDirect

Expert Systems with Applications

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

MCMR: Maximum coverage and minimum redundant text summarization model

Rasim M. Alguliev, Ramiz M. Aliguliyev *, Makrufa S. Hajirahimova, Chingiz A. Mehdiyev

Institute of Information Technology of National Academy of Sciences of Azerbaijan, Azerbaijan

ARTICLE INFO

Keywords:
Text summarization
Maximum coverage
Less redundancy
Integer linear programming
Particle swarm optimization
Branch-and-bound

ABSTRACT

In paper, we propose an unsupervised text summarization model which generates a summary by extracting salient sentences in given document(s). In particular, we model text summarization as an integer linear programming problem. One of the advantages of this model is that it can directly discover key sentences in the given document(s) and cover the main content of the original document(s). This model also guarantees that in the summary can not be multiple sentences that convey the same information. The proposed model is quite general and can also be used for single- and multi-document summarization. We implemented our model on multi-document summarization task. Experimental results on DUC2005 and DUC2007 datasets showed that our proposed approach outperforms the baseline systems.

1. Introduction

With the rapid development of information-communication technologies a huge amount of electronic documents has been produced and collected in the World Wide Web and digital libraries. According to 'How much information' the amount of electronic documents increases by 30% annually. This explosion of electronic documents has made it difficult for users to extract useful information from them. A lot of relevant and interesting documents are not read by the user due to the large amount of information.

To tackle this pressing text information overload problem, document clustering (Aliguliyev, 2009, 2009; Wang, Zhu, Li, Chi, & Gong, 2008) and text summarization (Aliguliyev, 2006, 2010; Alguliev & Alyguliev, 2008, 2009; Alguliev, Alyguliev, & Bagirov, 2005) together have been used as a solution. That is why document clustering enables us to group similar text information and then text summarization provides condensed text information for the similar text by extracting the most important text content from a similar document set or a document cluster. For this reason, document clustering and text summarization can be used for important components of information retrieval systems (Yoo, Hu, & Song, 2007). Present search engines usually provide a short summary for each retrieved document in order that users can quickly skim through the main content of the page. Therefore it saves users time and improves the search engine's service quality (Tao, Zhou, Lam, & Guan, 2008). That is why the necessity of tools that automatically generate summaries arises. These tools are not just for professionals who need to find the information in a short time but also for large searching engines such as Google, Yahoo!, AltaVista, and others, which could obtain a lot of benefits in its results if they use automatic generated summaries. After that, the user only will require the interesting documents, reducing the flow information (Garcia-Hernandez et al., 2009; Yang & Wang, 2008).

Automatic document summarization is one of the tasks that have long been studied in natural language processing. This task is the process of automatically creating a concise version of a text containing the main content of the original document (Kazantseva & Szpakowicz, 2010; Mani & Maybury, 1999). According to the type of information which they contain the summaries can be differently classified. For example, Tucker (1999) categorizes summaries as: *descriptive*, *evaluative*, *indicative* and *informative*. A *descriptive* summary can describe both the form and content of a source text. An *evaluative* summary offers some kind of critical response to the source, thereby evaluating it in some way. *Indicative* summaries give abbreviated information on the main topics of a document. They should preserve its most important passages and often used as the end part of the information retrieval systems, being returned by search system instead of full document. Thus, an indicative summary is aimed at helping the user to decide whether to read the information source, or not. By contrast, *informative* summaries provide a digest for full document, retaining important details, while reducing information volume (Fattah & Ren, 2009). The *summary type* factor is similar to the *style* output factor indicated by Jones (2007), who identifies three of the same types of summary as Tucker, *indicative*, *informative* and *critical*, and an additional one, *aggregative*, in which varied or multiple sources are summarized in relation to each other.

Depending on the number of documents to be summarized, the summary can be a single-document or a multi-document. Single-document summarization can only condense one document into a shorter representation, whereas multi-document summarization

* Corresponding author. Address: Institute of Information Technology of National Academy of Sciences of Azerbaijan, 9, F. Agayev Street, Baku AZ1141, Azerbaijan.

E-mail addresses: r.aliguliyev@gmail.com, a.ramiz@science.az, aramiz@iit.ab.az (R.M. Aliguliyev).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.05.033

can condense a set of documents into a summary. Multi-document summarization can be considered as an extension of single-document summarization and used for precisely describing the information contained in a cluster of documents and facilitate users to understand the document cluster. Since it combines and integrates the information across documents, it performs knowledge synthesis and knowledge discovery, and can be used for knowledge acquisition (Aliguliyev, 2010; Mani & Maybury, 1999; Radev, Blair-Goldensohn, & Zhang, 2001; Wan, 2008).

There are two approaches for document summarization: supervised and unsupervised (Tang, Yao, & Chen, 2009). The supervised approaches treat document summarization as a classification and the task formalize as identifying whether a sentence should be included in the summary or not. However, they require training samples. The unsupervised methods usually utilize clustering algorithms to score the sentences in the documents by combining a set of predefined features (Nomoto & Matsumoto, 2003; Yoo et al., 2007).

The summarization task can also be categorized as either generic or query-oriented. A query-oriented summary presents the information that is more relevant to the given queries, while a generic summary gives an overall sense of the document's content (Carbonell & Goldstein, 1998; Shen, Sun, & Li, 2007). In this paper, we focus on the unsupervised generic text summarization, which generates a summary by extracting key textual units in given document collection. Among other textual units that can be used in the method, we use sentences so that the grammaticality at the sentence level is going to be guaranteed. We represent generic text summarization model as an optimization problem and attempt to globally solve the problem. In particular, we model text summarization as an integer linear programming (ILP) problem. One of the advantages of this representation is that it can directly discover salient sentences in the given document(s) and cover the main content of the original source(s).

The rest of this paper is organized as follows. Section 2 introduces the brief review of the document summarization methods. The proposed generic text summarization model is presented in Section 3. Section 4 presents the algorithms to solve the ILP problem. The experiments and results are given in Section 5. Finally, we conclude our paper in Section 6.

2. Related work

Though text summarization has drawn attention primarily after the information explosion on the Internet, the first work has been done as early as in the 1950s (Luhn, 1958). Since then a variety of summarization methods has been proposed and evaluated. Generally, automatic document summarization methods can be divided into two categories: abstractive and extractive. In fact majority of researches have been focused on summary extraction, which selects the pieces such as keywords, sentences or even paragraph from the source to generate a summary. A human summarizer typically does not create a summary by extracting textual units verbatim from a source into the summary. Abstraction can be described as "reading and understanding the text to recognize its content which is then compiled in a concise text" (Kutlu, Cigir, & Cicekli, 2010). In general, an *abstract* can be described as a summary comprising concepts/ideas taken from the source which are then 'reinterpreted' and presented in a different form, whilst an *extract* is a summary consisting of units of text taken from the source and presented verbatim (Kutlu et al., 2010).

The extractive methods proposed in Alguliev and Aliguliyev (2005), Salton, Singhal, Mitra, and Buckley (1997) decompose a document in a set of paragraphs (sentences), using the cosine measure computes the similarity between paragraphs (sentences) and they represent the strength of the link between two paragraphs (sentences), and paragraphs (sentences) extracted according to different strategies.

In recent years, a variety of graph-based methods have been proposed for multi-document summarization (Erkan & Radev, 2004; Otterbacher, Erkan, & Radev, 2009; Radev et al., 2001; Wan & Xiao, 2009; Wan, Yang, & Xiao, 2007; Wei, Li, Lu, & He, 2008; Zhang, Cheng, Wu, & Xu, 2008, 2008; Zhao, Wu, & Huang, 2009). The graph-based methods first construct a graph representing the sentence relationships at different granularities and then evaluate the topic-biased saliency of the sentences based on the graph. Wan and Xiao (2009) considered the within-document relationships and the cross-document relationships between sentences as two separate modalities, and proposed to use the multi-modality manifold-ranking algorithm to fuse the two modalities. A transductive approach (Amini & Usunier, 2009) for extractive multi-document summarization identifies topic themes within a document collection, which help to identify two sets of relevant and irrelevant sentences to a question. It first defines a prior probability of relevance for every sentence using the set of keywords associated to a given question and then iteratively learns a scoring function which fits the prior probabilities, and also minimizes the number of irrelevant sentences scored above the relevant ones. In Haghighi and Vanderwende (2009) have been presented an exploration of content models for multi-document summarization and demonstrated that the use of structured topic models can benefit summarization quality as measured by automatic and manual metrics.

The paper (He, Shao, Li, Yang, & Ma, 2008) presents an approach based on estimation of content terms. In the process of estimating content-terms, it makes full use of the relevant feature and the information richness feature for assigning importance to each of them. With summary content terms being identified correctly, the candidate sentences are ranked and best sentences are selected to form the summary.

In automatic document summarization, the selection process of the distinct ideas included in the document is called diversity. The diversity is very important evidence serving to control the redundancy in the summarized text and produce more appropriate summary. Many approaches have been proposed for text summarization based on the diversity. The pioneer work for diversity based text summarization is MMR (maximal marginal relevance), it was introduced by Carbonell and Goldstein (1998). MMR maximizes marginal relevance in retrieval and summarization. The sentence with high maximal relevance means it is highly relevant to the query and less similar to the already selected sentences. The clustering plays an important role in text summarization (Alguliev & Alyguliev, 2008, 2009; Aliguliyev, 2006, 2009, 2010; Alguliev et al., 2005; Nomoto & Matsumoto, 2003). It is used as an effective tool for finding the diversity among the sentences. In Binwahlan, Salim, and Suanmali (2009) two ways were used for finding the diversity: the first one is a preliminary way where the document sentences are clustered based on the similarity and all resulting clusters are presented as a tree containing a binary tree for each group of similar sentences. The second way is to apply the proposed method on each branch in the tree to select one sentence as summary sentence. The clustering algorithm and binary tree were used as helping factor for finding the most distinct topics in the text. Nomoto and Matsumoto presented a new unsupervised approach for text summarization where evaluation does not rely on matching extracts against human-made summaries but measuring the loss of information in extracts in terms of retrieval performance. This approach firstly clusters the sentences and uses the obtained sentence clusters to generate a summary. The University of Michigan's summarization system, named MEAD, was initially developed to produce multi-document extractive summaries. The main idea behind MEAD is the use of the centroid-based feature which identifies sentences that are highly relevant to an entire cluster of related documents. For each sentence, MEAD then computes three values: the centroid score which is a measure of the centrality of a sentence to the overall topic of a cluster (or document in the case of a single-document cluster), the position score which decreases linearly as the sentence gets farther from the beginning of a document, and the overlap-with-first score which is the inner product of the tf-idf (term frequency-inverse document frequency) weighted vector representations of a given sentence and the first sentence (or title, if there is one) of the document (Radev et al., 2001). Recently, a new language model, factorization with given bases (FGB) (Wang et al., 2008) is proposed for document clustering and summarization by making use of both word-document matrix and word-sentence matrix.

In Yang and Wang (2008), a novel summarization model based on fractal theory has been presented. This model creates the summary of a document by a recursive deterministic algorithm based on the hierarchical document structure. The original document is represented as a fractal tree according to its document structure. The system extracts the sentences from the top level to the lower levels.

Paper (Lee, Park, Ahn, & Kim, 2009) presents a novel generic document summarization method using the generic relevance of a sentence based on negative matrix factorization (NMF). The proposed method has the following advantages: NMF selects more meaningful sentences than the LSA-related methods, because it can use more intuitively interpretable semantic features and grasp the innate structure of documents. Wang, Zhu, Li, and Gong (2009) proposed a Bayesian sentence-based topic model (BSTM) for multi-document summarization by making use of both the word-document and word-sentence associations. The BSTM models the probability distributions of selecting sentences given topics and provides a principled way for the summarization task. Tao et al. (2008) have designed word-based and sentence-based association networks (WAN and SAN for short, respectively) and proposed word and sentence weighting approaches based on how much co-occurrence information they contain, and applied to text summarization.

Filatova and Hatzivassiloglou (2004) modeled extractive document summarization as a maximum coverage problem that aims at covering as many conceptual units as possible by selecting some sentences. McDonald (2007) formalized text summarization as a knapsack problem and obtained the global solution and its approximate solutions. In Takamura and Okumura (2009), Takamura and Okamura represented text summarization as maximum coverage problem with knapsack constraint. But in Shen et al. (2007) they represented document summarization as a sequential labeling task and it solved with conditional random fields. Although this task is globally optimized in terms likelihood, the coverage of concepts is not taken into account. In Takamura and Okumura (2009), text summarization formalized as a budgeted median problem. This model covers the whole document cluster through sentence assignment, since in this model every sentence is represented by one of the selected sentences as much as possible. An advantage of this method is that it can incorporate asymmetric relations between sentences in a natural manner.

3. The proposed text summarization model

Assuming that the summarization task is to find the subset of sentences in text which in some way represents main content of source text, then arises a natural question: 'what are the properties of text that should be represented or retained in a summary'. A summary will be considered good, if the summary represents the whole content of the document(s) (Nomoto & Matsumoto, 2003). When creating summary from document(s), systems generally attempt to optimize three properties, namely

- *Relevance*: Summary should contain informative textual units that are relevant to the user.
- *Redundancy*: Summaries should not contain multiple textual units that convey the same information.
- *Length*: Summary is bounded in length.

Optimizing all three properties jointly is a challenging task and is an example of a global summarization problem. That is why the inclusion of relevant textual units relies not only on properties of the units themselves, but also properties of every other textual unit in the summary (McDonald, 2007).

3.1. Mathematical formalization

We would like to generate a summary such that similarity between a document collection and a summary is maximized. In the following, we introduce model for that purpose.

As input we are given a document collection $D = \{d_1, d_2, \ldots, d_{|D|}\}$, where $|D|$ is the number of documents. Each document d_i contains a set of sentences $d_i = \{s_1, s_2, \ldots, s_{d_{|i|}}\}$, $|d_i|$ is the number of sentences in the d_i, $i = 1, \ldots, |D|$. For simplicity, we represent the document collection simply as the set of all sentences from all the documents in the collection, i.e. $D = \{s_1, s_2, \ldots, s_n\}$, where s_i denotes ith sentence in D, n is the number of sentences in the document collection, $s_i \in D$ iff $s_j \in d_i \in D$. Let $T = \{t_1, t_2, \ldots, t_m\}$ represents all the terms occurring in D, where m is the number of different terms. We attempt to find a subset of the sentences $D = \{s_1, s_2, \ldots, s_n\}$ that covers the main content of the document collection. If we let $S \in D$ be the set of sentences constituting a summary, then the similarity between the document collection and the summary is going to be $sim(\vec{D}, \vec{S})$, which we would like to maximize. Here $\vec{D}$ denotes a feature vector of the document collection, $\vec{S}$ denotes a feature vector of the summary S, and $sim(\vec{D}, \vec{S})$ denotes the similarity of two vectors $\vec{D}$ and $\vec{S}$. In our study we use the cosine similarity and the NGD-based similarity measure because summary to the document or the entire document cluster are supposed to be important in summarization (Aliguliyev, 2009, 2010; Alguliev & Alyguliev, 2008, 2009). These similarity measures will be defined below.

We next have to impose the cardinality constraint on this maximization so that we can obtain a summary of length L or shorter. The length L is measured, for example, by the number of words or bytes in the summary.

Formally we can formalize the document summarization problem as follows:

$$\text{maximize } sim(\vec{D}, \vec{S}), \tag{1}$$

$$\text{s.t.} \quad len(S) \leqslant L, \tag{2}$$

where $len(S)$ denotes the length of the summary $S \in D$.

Apparently the problem (1) and (2) does not guarantee that a summary will not contain multiple sentences that convey the same information, i.e. redundancy in a summary will not be minimized. On the other hand, from the computational complexity viewpoint such formalization is not effective as at each step the feature vector of the summary should be calculated anew. To overcome these drawbacks we formalize the problem (1) and (2) in another way.

We introduce the following notations. Let x_{ij} denotes a variable which is 1 if pair of sentences s_i and s_j are selected, to be included in the summary, otherwise 0, and $len(s_i)$ denotes the length of sentence s_i.

Thus, assuming that each sentence is a candidate-summary sentence, then the problem (1) and (2) we can rewrite as:

$$\text{maximize} f = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left[sim\left(\vec{\mathrm{D}},\vec{s}_i\right)+sim\left(\vec{\mathrm{D}},\vec{s}_j\right)-sim\left(\vec{s}_i,\vec{s}_j\right)\right]x_{ij}, \quad (3)$$

$$\text{s.t.} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}[len(s_i)+len(s_j)]x_{ij} \leqslant L, \quad (4)$$

$$x_{ij} \in \{0,1\}, \ \forall i,j, \quad (5)$$

where $\vec{s}_i$ denotes a feature vector of the sentence s_i.

Now our objective is to find the binary assignment on x_{ij} (5) with the best coverage and less redundancy (3) such that the summary length is at most L (4). Eqs. (3)–(5) is an integer linear programming (ILP) problem, where both the objective function (3) and the constraint (4) are linear in the set of integer variables (5). The objective function (3) guarantees that the main content of the document collection will be covered by the summary, i.e. the summary will be relevance to the user (this is guaranteed by the first and second terms). This function also guarantees that in the summary will not be multiple textual units that convey the same information, i.e. redundancy in the summary will be reduced (this provides the third term). Eq. (4) is the cardinality constraint, which guarantees that the summary is bounded in length. The integrality constraint on x_{ij} (5) is automatically satisfied in the problem above.

3.2. Similarity measure

For calculation of similarity between textual units, each of them should be presented as a vector. The vector space model is the most known representation scheme for textual units. The vector space model represents textual units by counting terms or sequence of terms.

3.2.1. Cosine similarity

It uses the weighting terms representation of the textual units. According to this representation the sentence s_i represented as a weighting vector of the terms, $\vec{s}_i = \{w_{i1}, w_{i2}, \ldots, w_{im}\}$, where m is the number of the terms in the document collection, w_{ik} is the weight of the term t_k in the sentence s_i. The component w_{ik} is defined using the scheme *tfisf*. This scheme combines the definitions of term frequency and inverse sentence frequency, to produce a composite weight for each term in each sentence. This weighting scheme assigns to term a weight in sentence given by

$$w_{ik} = f_{ik} \times \log\left(\frac{n}{n_k}\right), \quad (6)$$

where f_{ik} is the term frequency (i.e. denotes how many term t_k occurs in sentence s_i), n_k denotes the number of sentences in which term t_k appears. The term log (n/n_k), which is very often referred to as the *isf* factor, accounts for the global weighting of term t_k. The *isf* factor has been introduced to improve the discriminating power of terms in the traditional information retrieval.

In other words, *tf-isf* assigns to term t_k a weight in sentence s_i that is

- highest when t_k occurs many times within a small number of sentences;
- lower when the term occurs fewer times in a sentence, or occurs in many sentences;
- lowest when the term occurs in virtually all sentences.

Then, the cosine similarity between two vectors $\vec{s}_i$ and $\vec{s}_i$ we calculate as

$$sim_{\cos}(\vec{s}_i,\vec{s}_j) = \frac{\sum_{k=1}^{m} w_{ik}w_{jk}}{\sqrt{\sum_{k=1}^{m} w_{ik}^2 \cdot \sum_{k=1}^{m} w_{jk}^2}}, \quad i,j = 1,\ldots,n. \quad (7)$$

3.2.2. NGD-based similarity

To calculate the NGD-based similarity between two sentences each of them must be represented as a sequence of terms (Aliguliyev, 2009, 2010; Alguliev & Aliguliyev, 2009). For example, the sentence s_i is represented as a set of distinct terms appearing in it, $s_i = \{t_1, t_2, \ldots, t_{|s_i|}\}$, where $|s_i|$ is the number of distinct terms in s_i. Then the similarity between two sentences s_i and s_j we calculate as follows (Alguliev & Aliguliyev, 2009):

$$sim_{\mathrm{NGD}}(s_i,s_j) = \frac{\sum_{t_k\in s_i}\sum_{t_l\in s_j} sim_{\mathrm{NGD}}(t_k,t_l)}{|s_i|\cdot|s_j|}, \quad (8)$$

where

$$sim_{\mathrm{NGD}}(t_k,t_l) = \exp(-\mathrm{NGD}(t_k,t_l)) \quad (9)$$

– is the NGD-based similarity measure between terms t_k and t_l .

In (9)NGD (t_k,t_l) is the Normalized Google Distance (Cilibrasi & Vitanyi, 2007) between terms t_k and t_l, which in Aliguliyev (2009, 2010), Alguliev and Alyguliev (2009) is slightly modified:

$$\mathrm{NGD}(t_k,t_l) = \frac{\max\{\log(f_k),\log(f_l)\} - \log(f_{kl})}{\log n - \min\{\log(f_k),\log(f_l)\}}, \quad (10)$$

where f_k is the number of sentences containing the term t_k, f_{kl} denotes the number of sentences containing both terms t_k and t_l, n is the number of sentences in the document collection.

3.3. Weighted combination of the objective functions

As it is known (Aliguliyev, 2010), the similarity measure plays an important role in text summarization. In order to investigate how the relative contributions from the cosine measure and the NGD-based measure between sentences influence the summarization performance, we propose to define the final objective function by linearly combing the objective function based on the cosine similarity measure and the objective function based on the NGD-based similarity measure as follows:

$$\text{maximize } f_\alpha = \alpha \cdot f_{\cos} + (1-\alpha)\cdot f_{\mathrm{NGD}}, \quad (11)$$

where

$$f_{\cos} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}[sim_{\cos}(\vec{\mathrm{D}},\vec{s}_i)+sim_{\cos}(\vec{\mathrm{D}},\vec{s}_j)-sim_{\cos}(\vec{s}_i,\vec{s}_j)]x_{ij}, \quad (12)$$

$$f_{\mathrm{NGD}} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}[sim_{\mathrm{NGD}}(\vec{\mathrm{D}},\vec{s}_i)+sim_{\mathrm{NGD}}(\vec{\mathrm{D}},\vec{s}_j)-sim_{\mathrm{NGD}}(\vec{s}_i,\vec{s}_j)]x_{ij}. \quad (13)$$

The weighting parameter $\alpha \in [0,1]$ specifies the relative contributions to the final information richness of sentences from the cosine similarity measure and the NGD-based measure between sentences. If $\alpha = 0$, f_α is equal to f_{NGD}; if $\alpha = 0$, f_α is equal to $f_{\cos}$; and $\alpha = 0.5$, the cosine measure and the NGD-based measure are assumed to be equally important.

4. Algorithms for solving the optimization problem

In this section, we explain the algorithms applied to solve the ILP problem (11), (4) and (5).

4.1. Branch-and-bound algorithm

Solving arbitrary ILPs is an NP-hard problem. However, ILPs are a well studied optimization problem with efficient branch-and-bound (B&B) algorithms for finding the optimal solution. Since our model is an NP-hard problem, it cannot generally be solved in polynomial time. However, if the size of the problem is limited, sometimes we can obtain the exact solution within a practical time by means of the branch-and-bound method. Modern commercial ILP solvers can typically solve moderately large optimizations in a matter of seconds. We use the GNU Linear Programming kit (GNU Linear Programming, XXXX), which is a free optimization package.

4.2. Binary particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm is a stochastic population-based search algorithm inspired by the social behavior of bird flocks or schools of fish. Since its introduction in 1995, PSO has drawn much attention and has been applied to the solution of optimization problems (Poli, Kennedy, & Blackwell, 2007).

The PSO algorithm for a global optimization problem uses a swarm of N_{sw} particles. Each particle i of the swarm is associated with a position in the continuous n dimensional search space $p_i(t) = [p_{i1}(t),\dots,p_{in}(t)]$, $i = 1,\dots,N_{sw}$. For each particle i, the best previous position (*personal best*), and the particle position change (*velocity*) are recorded and represented respectively as $p_i^{best} = [p_{i1}^{best},\dots,p_{in}^{best}]$ and $v_i(t) = [v_{i1}(t),\dots,v_{in}(t)]$. The position associated with the current best function value is denoted as $g^{best} = [g_1^{best},\dots,g_n^{best}]$ (*global best*).

The personal best position of particle x_i at iteration $(t+1)$ is calculated as:

$$p_i^{best}(t+1) = \begin{cases} p_i^{best}(t), & \text{if } f_\alpha(p_i(t+1)) \leqslant f_\alpha(p_i^{best}(t)), \\ p_i(t+1), & \text{if } f_\alpha(p_i(t+1)) > f_\alpha(p_i^{best}(t)), \end{cases} \tag{14}$$

where $f_\alpha(\cdot)$ is the objective function (11).

At the iteration t the best position of swarm is computed as:

$$g^{best}(t) = \max\{f_\alpha(p_1(t)),\dots,f_\alpha(p_{N_{sw}}(t))\}. \tag{15}$$

The following equations are used to update iteratively the particles' velocities and positions:

$$v_{ij}(t+1) = w\cdot v_{ij}(t) + c_1\cdot r_1\cdot(p_{ij}^{best}(t) - p_{ij}(t)) + c_2\cdot r_2\cdot(g_j^{best}(t) - p_{ij}(t)), \tag{16}$$

$$p_{ij}(t+1) = p_{ij}(t) + v_{ij}(t+1), \tag{17}$$

where $p_{ij}(t)$ and $v_{ij}(t)$ denote respectively the position and velocity of the ith particle with respect to jth dimension ($j = 1,2,\dots,n$) at iteration t, $t = 0,1,\dots$, indicates the iteration number, r_1 and r_2 are two independent random numbers uniformly distributed within the interval $[0,1]$.

The initial value of the position and the velocity vectors are generated as uniform random variables by the following rules:

$$p_{ij}(0) = p_{\min} + (p_{\max} - p_{\min})\cdot r_1, \tag{18}$$

$$v_{ij}(0) = v_{\min} + (v_{\max} - v_{\min})\cdot r_2, \tag{19}$$

where $v_{\min}$, $v_{\max}$ are the allowable minimum and maximum velocity values, $p_{\min}$, $p_{\max}$ are the allowable minimum position and maximum position of particles.

The inertia weight, w, controls the influence of previous velocity on the current velocity. Typically w is reduced linearly, from $w_{\max}$ to $w_{\min}$, each iteration, a good starting point is to set $w_{\max}$ to 0.9 and $w_{\min}$ to 0.4

$$w(t) = \frac{(t_{\max} - t)\cdot(w_{\max} - w_{\min})}{t_{\max}} + w_{\min}, \tag{20}$$

where t is the current iteration and $t_{\max}$ is the maximum number of iterations, $w_{\max}$ and $w_{\min}$ represent the starting and ending inertia weight values to control the inertia, respectively.

It was shown that a good convergence can be ensured by making the acceleration and inertia constants dependent. Their relation is shown in the following equation with an intermediate parameter φ

$$w = \frac{1}{\varphi - 1 + \sqrt{\varphi^2 - 2\varphi}}, \tag{21}$$

$$c_1 = c_2 = \varphi\cdot w. \tag{22}$$

The swarm size N_{sw} plays an important role in evolutionary methods. An appropriate population size can maintain the effectiveness of the algorithm. It is quite a common practice in the PSO literature to limit the number of particles to the range from 20 to 60.

The binary PSO algorithm was introduced to allow the PSO algorithm to operate in binary problem spaces. The major difference between binary PSO with continuous version is that velocities of the particles are rather defined in terms of probabilities that a bit will change to one. Using this definition a velocity must be restricted within the range $[0,1]$. So a map is introduced to map all real valued numbers of velocity to the range $[0,1]$. In the binary PSO, Eq. (16) for updating the velocity remains unchanged, but Eq. (16) for updating the position is redefined by the rule:

$$p_{ij}(t+1) = \begin{cases} 1, & \text{if } \mathrm{rand}_j < sigm(v_{ij}(t+1)), \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

where rand_j is the random number selected from a uniform over $[0,1]$, and $sigm(\cdot)$ is the sigmoid function for transforming the velocity to the probability as the following expression

$$sigm(v_{ij}(t+1)) = \frac{1}{1 + \exp(-v_{ij}(t+1))}. \tag{24}$$

The complete computational procedure of the binary PSO can be summarized as follows:

Step 1 (Initialize). Initialize parameters and population with random position and velocities using Eqs. (18)–(22).
Step 2 (Evaluation). Evaluate the fitness value (the desired objective function) for each particle.
Step 3 (Find the personal best). If the fitness value of particle i is better than its best fitness value (*personal best*), then set the current fitness value as the new *personal best* to particle i (Eq. (14)).
Step 4 (Find global best). If any *personal best* is updated and it is better than the current *global best*, then set *global best* to the current value (Eq. (15)).
Step 5 (Update velocity and position). Update velocity and move to the next position according Eqs. (16) and (23).
Step 6 (Stopping criterion). If the maximum number of iterations, $t_{\max}$, is met, then stop; otherwise go back to step 2.

5. Experiments

In this section, we conducted the experiments to evaluate the performance of the proposed method.

5.1. Datasets

Following the most relevant previous methods we evaluated the proposed model on the DUC2005 and DUC2007 datasets, both of which are open benchmark datasets from Document Understanding Conference (DUC) (Document Understanding Conference, XXXX) for automatic summarization evaluation. DUC2005 and DUC2007 datasets consist of 50 and 45 topics, respectively. Each topic of DUC2005 includes 25 $\sim$ 50 documents and each topic of DUC2007 includes a fixed number – 25 documents. The task is to create a summary of no more than 250 words for each topic to answer the information expressed in the topic statement. Table 1 gives a brief description of the datasets.

All the documents were segmented into sentences using a script distributed by DUC. For similarity computation, we preprocessed each sentence by (a) removing stopwords; (b) removing words that appear less than five times in the corpus; and stemming the remaining words. For removing the stopwords we used the stoplist from English stoplist (XXXX), which contains about 600 common words. In our experiments, stopwords were stemmed using the Porter's scheme (Porter Stemming Algorithm, XXXX).

5.2. Performance evaluation metrics

For performance evaluation we used the ROUGE-1.5.5 (Recall-Oriented Understudy for Gisting Evaluation) package (Lin, 2004). ROUGE is adopted by DUC as the official evaluation metric for text summarization. It includes measures which automatically determine the quality of a machine summary by comparing it to other (ideal) summaries created by humans: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. These measures evaluate the quality of the summarization by counting the number of overlapping units, such *N*-grams, between the generated summary by a method and a set of reference summaries.

Basically, the ROUGE-N measure compares *N*-grams of two summaries, and counts the number of matches. This measure is computed as Lin (2004):

$$ROUGE\text{-}N = \frac{\sum_{S \in Summ_{ref}} \sum_{N\text{-gram} \in S} Count_{match}(N\text{-gram})}{\sum_{S \in Summ_{ref}} \sum_{N\text{-gram} \in S} Count(N\text{-gram})}, \quad (25)$$

where N stands for the length of the N-gram, $Count_{match}$ (N-$gram$) is the maximum number of N-grams co-occurring in candidate summary and the set of reference-summaries. $Count$(N-$gram$) is the number of N-grams in the reference summaries.

N-gram overlap with $N = 1$ behaves similarly to cosine similarity. But for $N > 1$, N-gram overlap is a more strict matching algorithm than cosine similarity, because it is sensitive to the ordering of words in a sentence.

Lin (2004) implemented two extensions to ROUGE-N: skip-bigram co-occurrence (ROUGE-S) and skip-bigram co-occurrence averaged with unigram co-occurrence (ROUGE-SU). The way ROUGE-S is calculated identical to ROUGE-2, except that skip bigrams are defined as *subsequences* rather than the regular definition of bigrams as *substrings*. Skip-bigram (skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps) co-occurrence statistics, ROUGE-S, measure the similarity of a pair of summaries based on how many skip-bigrams they have in common:

$$P_{SKIP2}(R,S) = \frac{SKIP2(R,S)}{C(|S|,2)}, \quad R_{SKIP2}(R,S) = \frac{SKIP2(R,S)}{C(|R|,2)},$$
$$F_{SKIP2}(R,S) = \frac{(1+\beta^2)P_{SKIP2}(R,S)R_{SKIP2}(R,S)}{\beta^2 P_{SKIP2}(R,S) + R_{SKIP2}(R,S)}, \quad (26)$$

where $SKIP2(R,S)$ is the number of skip-bigram matches between R and S, β is the relative importance of $P_{SKIP2}(R,S)$ and $R_{SKIP2}(R,S)$, $P_{SKIP2}(R,S)$ being the precision of $SKIP2(R,S)$ and $R_{SKIP2}(R,S)$ a recall of $SKIP2(R,S)$. C is the combination function.

Table 1
Description of the datasets.

	DUC 2005	DUC 2007
Number of clusters	50	45
Number of documents	1593	1125
Data source	TREC	AQUAINT
Summary length	250 words	250 words

One potential problem for ROUGE-S is that it does not give any credit to a candidate sentence if the sentence does not have any word pair co-occurring with its references. To accommodate this, we extend ROUGE-S with the addition of unigram as counting unit. ROUGE-SU is an extension of ROUGE-S with the addition of unigram as the counting unit, which is a weighted average between ROUGE-S and ROUGE-1.

For evaluation of our method, we use the ROUGE-2 and ROUGE-SU4 metrics. ROUGE-2 compares the bigram overlap between the system summary and the manual summaries created by human. ROUGE-SU4 is an extended version of ROUGE-2 that allows word-level gaps of maximum length 4 between the bigram tokens.

5.3. Results and analysis

We compared our method on DUC2005 dataset with the six methods: TranSumm (Amini & Usunier, 2009), QEA (Zhao et al., 2009), Content-term (He et al., 2008), Biased LexRank (Otterbacher et al., 2009), TMR + TF (Tang et al., 2009) and Qs-MRC (Wei et al., 2008). For evaluation of our method on DUC2007 dataset we selected the following methods: PNR^2(Wenjie, Furu, Qin, & Yanxiang, 2008), PPRSum (Liu, Wang, Zhang, & Xu, 2008), GSPSum (Zhang, Xu, & Cheng, 2008) and AdaSum (Zhang et al., 2008). These methods have been chosen for comparison because they have achieved the best results on the DUC2005 and DUC2007 datasets. The experimental results are shown in Tables 2 and 3. In Tables 2 and 3 MCMR (Maximum Coverage and Minimum Redundant) denotes the proposed method (in brackets by the B&B and PSO respectively denoted the branch-and-bound and particle swarm optimization algorithms which have been used for solving the optimization problem).

The PSO algorithm is stochastic in nature. Hence, it has been run several times. The parameters of the binary PSO are set as follows: the swarm size, $N_{sw} = 30$; the number of iteration, $t_{\max} = 500$; the allowable minimum position and maximum position of particles, $p_{\max} = 50$; $p_{\min} = -50$; the allowable minimum and maximum velocity values, $v_{\max} = 5$; $v_{\min} = -5$. In the experiments, the weighting parameter α is set as follows: $\alpha = 0.65$ (on the DUC2005 dataset) and $\alpha = 0.55$ (on the DUC2007 dataset). The impact of using different α's are further studied.

With comparison to the average ROUGE values for other methods, our method can achieve significant improvement. Results of comparison reported in Tables 4 and 5. We observe that the result of our method directly depends on the optimization algorithm. As shown in Tables 4 and 5, among two algorithms B&B and PSO the best result obtained by the B&B. We observe also that our method MCMR (B&B) with the B&B optimization algorithm demonstrates the best ROUGE values and outperforms all the other systems on both datasets. Among other methods the best results have been shown by the Qs-MRC and PPRSum methods on DUC2005 and DUC2007 datasets, respectively. Comparison with the method Qs-MRC on DUC2005 dataset shows that our method MCMR

Table 2
ROUGE values of the methods on DUC2005 dataset.

Methods	Source	ROUGE-2	ROUGE-SU4
MCMR (B&B)		**0.0790**	**0.1392**
MCMR (PSO)		0.0754	0.1360
TranSumm	Amini and Usunier (2009)	0.0755	0.1366
QEA	Zhao et al. (2009)	0.0749	0.1333
Content-term	He et al. (2008)	0.0718	0.1338
Biased LexRank	Otterbacher et al. (2009)	0.0753	0.1363
Qs-MRC	Wei et al. (2008)	0.0779	0.1366
TMR + TF	Tang et al. (2009)	0.0715	0.1304

Table 3
ROUGE values of the methods on DUC2007 dataset.

Methods	Source	ROUGE-2	ROUGE-SU4
MCMR (B&B)		**0.1221**	**0.1753**
MCMR (PSO)		0.1165	0.1697
PNR^2	Wenjie et al. (2008)	0.0895	0.1291
PPRSum	Liu et al. (2008)	0.1195	0.1710
GSPSum	Zhang et al. (2008)	0.1110	0.1638
AdaSum	Zhang et al. (2008)	0.1172	0.1692

Table 4
Comparison MCMR (B&B) with other methods on DUC2005 dataset.

Methods	Improvement of the method MCMR (B&B) (%)	
	ROUGE-2	ROUGE-SU4
MCMR (B&B)	0.00	0.00
MCMR (PSO)	4.77	2.35
TranSumm	4.64	1.90
QEA	5.47	4.43
Content-term	10.03	4.04
Biased LexRank	4.91	2.13
Qs-MRC	1.41	1.90
TMR + TF	10.49	6.75

Table 5
Comparison MCMR with other methods on DUC2007 dataset.

Methods	Improvement of the method MCMR (B&B) (%)	
	ROUGE-2	ROUGE-SU4
MCMR (B&B)	0.00	0.00
MCMR (PSO)	4.81	3.30
PNR^2	36.42	35.79
PPRSum	2.18	2.51
GSPSum	10.00	7.02
AdaSum	4.18	3.61

Table 6
Influence of parameter α to ROUGE-2 value of the method f_α.

Parameter, α	DUC2005	DUC2007
0.00	0.0781	0.1185
0.05	0.0781	0.1186
0.10	0.0782	0.1189
0.15	0.0783	0.1194
0.20	0.0783	0.1201
0.25	0.0784	0.1208
0.30	0.0784	0.1209
0.35	0.0784	0.1211
0.40	0.0785	0.1213
0.45	0.0786	0.1217
0.50	0.0787	0.1216
0.55	0.0788	**0.1221**
0.60	0.0789	0.1218
0.65	**0.0790**	0.1216
0.70	0.0788	0.1215
0.75	0.0787	0.1213
0.80	0.0787	0.1211
0.85	0.0786	0.1209
0.90	0.0784	0.1209
0.95	0.0783	0.1208
1.00	0.0783	0.1207

(B&B) improves the performance by 1.41% and 1.90% in terms ROUGE-2 and ROUGE-SU4 metrics, respectively. Comparison also with the PPRSum on DUC2007 dataset shows the method MCMR (B& B) improves the performance by 2.18% and 2.51% in terms ROUGE-2 and ROUGE-SU4 metrics, respectively. Here for comparison we have used the relative improvement $\frac{(\text{ourmethod}-\text{othermethods})}{\text{othermethods}} \times 100$ for comparison. In Tables 2, 3, 6 and 7 bold entries represent the best performing summarization methods in terms of evaluation metrics. In these tables through MCMR (Maximum Coverage and Minimum Redundant) denoted our model with the objective function f_α.

Table 7
Influence of parameter α to ROUGE-SU4 value of the method f_α.

Parameter, α	DUC2005	DUC2007
0.00	0.1366	0.1716
0.05	0.1367	0.1721
0.10	0.1368	0.1726
0.15	0.1368	0.1728
0.20	0.1370	0.1733
0.25	0.1372	0.1737
0.30	0.1373	0.1740
0.35	0.1375	0.1742
0.40	0.1378	0.1745
0.45	0.1382	0.1747
0.50	0.1385	0.1750
0.55	0.1388	**0.1753**
0.60	0.1388	0.1749
0.65	**0.1392**	0.1746
0.70	0.1389	0.1742
0.75	0.1386	0.1738
0.80	0.1381	0.1736
0.85	0.1378	0.1731
0.90	0.1375	0.1727
0.95	0.1372	0.1722
1.00	0.1371	0.1721

In this section, we also show the evaluation results of the objective function f_α (11) under the different values of the parameter α. In order to investigate the influences of the parameter α in the proposed method (11), the parameter value of α is varied from 0 to 1. Tables 6 and 7 demonstrate the influence of the weighting parameter α to performance of the method f_α in ROUGE2 and ROUGE-SU4 scores, respectively.

The value $\alpha = 0$ corresponds to the function f_{NGD} (13), and $\alpha = 1$ represents the f_{cos}(12). We can see that when α varies between *0* and *1*, better summarization performances were observed for all ROUGE scores compared to that with $\alpha = 0$ and $\alpha = 1$. It means that use of weighted combination f_α leads to better performance than f_{cos} and f_{NGD}. We observe also that the objective function f_{cos} demonstrates better results than the objective function f_{NGD}, and each of them outperforms other baseline systems. From Tables 6 and 7, we observe that $\alpha = 0.65$ is the optimal value on DUC2005 dataset, and $\alpha = 0.55$ is the optimal value on DUC2007 dataset. Notice that, the ROUGE values reported in Tables 6 and 7 are obtained by the branch-and-bound algorithm.

6. Conclusion

We have presented an approach to automatic document. Our approach modeled as an integer linear programming problem. This model generally attempts to optimize three properties, namely, (1) relevance: summary should contain informative textual units that are relevant to the user; (2) redundancy: summaries should not contain multiple textual units that convey the same information; and (3) length: summary is bounded in length. The approach proposed in this paper is applicable to both tasks: single- and multi-document summarization. In both tasks, documents are split into sentences in preprocessing. We select some salient sentences from document(s) to generate a summary. Finally, the summary is generated by threading all the selected sentences in the order that they appear in the original document(s). Here, we implemented our model on multi-document summarization task. When comparing our methods to several existing summarization methods on an open DUC2005 and DUC2007 datasets, we found that our method can improve the summarization results significantly. The methods were evaluated using ROUGE-2 and ROUGE-SU4 metrics. In this paper we also demonstrated that the summarization result depends on the similarity measure. Results of experiment showed that combination of the NGD-based and cosine similarity measures conducts to better result than their use separately.

References

Alguliev, R.M., & Aliguliyev, R.M. (2005). Effective summarization method of text documents. In *Proceedings of the 2005 IEEE/WIC/ACM international conference on web intelligence (WI'05), Compiegne, France, September 19–22* (pp.2640–271).

Alguliev, R. M., & Alyguliev, R. M. (2008). Automatic text documents summarization through sentences clustering. *Journal of Automation and Information Sciences, 40*(9), 53–63.

Alguliev, R. M., & Aliguliyev, R. M. (2009). Evolutionary algorithm for extractive text summarization. *Journal of Intelligent Information Management, 1*(2), 128–138.

Alguliev, R. M., Alyguliev, R. M., & Bagirov, A. M. (2005). Global optimization in the summarization of text documents. *Automatic Control and Computer Sciences, 39*(6), 42–47.

Aliguliyev, R. M. (2006). A novel partitioning-based clustering method and generic document summarization. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT 2006 Workshops) (WI-IATW'06), Hong Kong, China, December 18–22* (pp. 626–629).

Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications, 36*(4), 7764–7772.

Aliguliyev, R. M. (2009). Performance evaluation of density-based clustering methods. *Information Sciences, 179*(20), 3583–3602.

Aliguliyev, R. M. (2009). Clustering of document collection – a weighting approach. *Expert Systems with Applications, 36*(4), 7904–7916.

Aliguliyev, R. M. (2010). Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization. *Computational Intelligence, 26*(4), 420–448.

Amini, M-R., & Usunier, N. (2009). Incorporating prior knowledge into a transductive ranking algorithm for multi-document summarization. In *Proceedings of the 32nd annual ACM SIGIR conference on research and development in information retrieval (SIGIR'09), Boston, USA, July 19–23* (pp. 704–705).

Binwahlan, M. S., Salim, N., & Suanmali, L. (2009). MMI diversity based text summarization. *International Journal of Computer Science and Security, 3*(1), 23–33.

Carbonell, J., & Goldstein, J. (1998). In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, Melbourne, Australia, August 24–28* (pp. 335–336).

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Transaction on Knowledge and Data Engineering, 19*(3), 370–383.

Erkan, G., & Radev, D. (2004). Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22*, 457–479.

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech and Language, 23*(1), 126–144.

Filatova, E., & Hatzivassiloglou, V. (2004). A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on computational linguistics (COLING'04), Geneva, Switzerland, August 23–27* (pp.397–403).

Garcia-Hernandez, R. A., Ledeneva, Y., & Mendoza G. M. et al. (2009). Comparing commercial tools and state-of-the-art methods for generating text summaries. In *Proceedings of the eighth Mexican international conference on artificial intelligence (MICAI 2009), Guanajuato, Mexico, November 9–13* (pp. 92–96).

Haghighi, A., & Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics (NAACL'09), Boulder, Colorado, May 31–June 05* (pp.362–370).

He, T., Shao, W., Li, F., Yang, Z., & Ma, L. (2008). The automated estimation of content-terms for query-focused multi-document summarization. In *Proceedings of the 2008 fifth international conference on fuzzy systems and knowledge discovery (FSKD 2008), October 18–20, Jinan, China, vol. 5* (pp.580–584).

Jones, K. S. (2007). Automatic summarizing: The state of the art. *Information Processing and Management, 43*(6), 1449–1481.

Kazantseva, A., & Szpakowicz, S. (2010). Summarizing short stories. *Computational Linguistics, 36*(1), 71–109.

Kutlu, M., Cigir, C., & Cicekli, I. (2010). Generic text summarization for Turkish. *The Computer Journal, 53*(8), 1315–1323.

Lee, J-H., Park, S., Ahn, C-M., & Kim, D. (2009). Automatic generic document summarization based on non-negative matrix factorization. *Information Processing and Management, 45*(1), 20–34.

Lin, C-Y. (2004). ROUGE: A package for automatic evaluation summaries. In *Proceedings of the workshop on text summarization branches out, Barcelona, Spain, July 25–26* (pp. 74–81).

Liu, Y., Wang, X., Zhang, J., & Xu, H. (2008). Personalized PageRank based multi-document summarization. In *Proceedings of the first IEEE international workshop on semantic computing and systems (WSCS2008), Huangshan, China, July 14–15* (pp.169–173).

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159–165.

Mani, I., & Maybury, M. T. (1999). *Advances in automatic text summarization*. Cambridge: MIT Press. 442p.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of 29th European conference on IR research (ECIR 2007). LNCS*. Rome, Italy: Springer-Verlag. no.4425, pp. 557–564.

Nomoto, T., & Matsumoto, Y. (2003). The diversity-based approach to open-domain text summarization. *Information Processing and Management, 39*(3), 363–389.

Otterbacher, J., Erkan, G., & Radev, D. R. (2009). Biased LexRank: Passage retrieval using random walks with question-based priors. *Information Processing and Management, 45*(1), 42–54.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence, 1*(1), 33–57.

Radev, D. R., Blair-Goldensohn, S., & Zhang, Z. (2001). Experiments in single and multidocument summarization using MEAD. In *Proceedings of the first document understanding conference, New Orleans, USA, September 13–14* (p. 8).

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing and Management, 33*(2), 193–207.

Shen, D., Sun, J. -T., & Li, H. et al. (2007). Document summarization using conditional random fields. In *Proceedings of the 20th international joint conference on artificial intelligence (IJCAI-2007), Hyderabad, India, January 6–12* (pp. 2862–2867).

Takamura, H., & Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th conference of the European chapter of the ACL, Athens, Greece, March 30–April 3* (pp. 781–789).

Takamura, H., & Okumura, M. (2009). Text summarization model based on the budgeted median problem. In *Proceedings of the 18th ACM international conference on information and knowledge management (CIKM'09), Hong Kong, China, November 2–6* (pp.1589–1592).

Tang, J., Yao, L., & Chen, D. (2009). Multi-topic based query-oriented summarization. In *Proceedings of the ninth SIAM international conference on data mining, Nevada, USA, April 30–May 2* (pp. 1148–1159).

Tao, Y., Zhou, S., Lam, W., & Guan, J. (2008). Towards more text summarization based on textual association networks. In *Proceedings of the 2008 fourth international conference on semantics, knowledge and grid, Beijing, China, December 03–05* (pp.235–240).

Tucker, R. (1999). *Automatic summarizing and the CLASP system*. PhD thesis. University of Cambridge, UK, p. 190 <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-484.pdf>.

Wan, X. (2008). Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval, 11*(1), 25–49.

Wan, X., & Xiao, J. (2009). Graph-based multi-modality learning for topic-focused multi-document summarization. In *Proceedings of the 21st international joint conference on artificial intelligence (IJCAI'09), Pasadena, USA, July 11–17* (pp. 1586–1591).

Wan, X., Yang, J., & Xiao, J. (2007). Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of the 20th international joint conference on artificial intelligence (IJCAI-2007), Hyderabad, India, January 6–12* (pp. 2903–2908).

Wang, D., Zhu, S., Li, T., & Gong, Y. (2009). Multi-document summarization using sentence-based topic models. In *Proceedings of ACL-IJCNLP conference, Singapore, 4th August* (pp.297–300).

Wang, D., Zhu, S., Li, T., Chi, Y., & Gong, Y. (2008). Integrating clustering and multi-document summarization to improve document understanding. In *Proceedings of the ACM 17th conference on information and knowledge management (CIKM'08), Napa Valley, USA, October 26–30* (p.1435–1436).

Wei, F., Li, W., Lu, Q., & He, Y. (2008). Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'08), Singapore, July 20–24* (pp. 283–290).

Wenjie, L., Furu, W., Qin, L., & Yanxiang, H. (2008). PNR^2: Ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of the 22nd international conference on computational linguistics (COLING'08), Manchester, UK, August 18–22, vol.1* (pp. 489–496).

Yang, C. C., & Wang, F. L. (2008). Hierarchical summarization of large documents. *Journal of the American Society for Information Science and Technology, 59*(6), 887–902.

Yoo, I., Hu, X., & Song, I-Y. (2007). A coherent graph-based semantic clustering and summarization approach for biomedical literature and a new summarization evaluation method. *BMC Bioinformatics, 8* (Supp 9), S4, 15p.

Zhang, J., Cheng, X., Wu, G., & Xu, H. (2008). Ada Sum: An adaptive model for summarization. In *Proceedings of the ACM 17th conference on information and knowledge management (CIKM'08), Napa Valley, USA, October 26–30* (pp. 901–909).

Zhang, J., Xu, H., & Cheng, X. (2008). GSPSummary: A graph-based sub-topic partition algorithm for summarization. In *Proceedings of the 2008 Asia information retrieval symposium, Harbin, China, January 15–18* (pp. 321–334).

Zhao, L., Wu, L., & Huang, X. (2009). Using query expansion in graph-based approach for query-focused multi-document summarization. *Information Processing and Management, 45*(1), 35–41.
Document Understanding Conference: <http://duc.nist.gov>.
English stoplist: <ftp://ftp.cs.cornell.edu/pub/smart/english.stop>.
GNU Linear Programming: <http://www.gnu.org/software/glpk/>.
How much information: <http://www.sims.berkeley.edu./how-much-info-2003>.
Porter Stemming Algorithm: <http://www.tartarus.org/martin/PorterStemmer/>.