Contents lists available at ScienceDirect

Expert Systems with Applications

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

A new sentence similarity measure and sentence based extractive technique for automatic text summarization

Ramiz M. Aliguliyev *

Institute of Information Technology of National Academy of Sciences of Azerbaijan, 9, F.Agayev str., AZ1141 Baku, Azerbaijan

ARTICLE INFO

Keywords:
Similarity measure
Text mining
Sentence clustering
Summarization
Evolution algorithm
Sentence extractive technique

ABSTRACT

The technology of automatic document summarization is maturing and may provide a solution to the information overload problem. Nowadays, document summarization plays an important role in information retrieval. With a large volume of documents, presenting the user with a summary of each document greatly facilitates the task of finding the desired documents. Document summarization is a process of automatically creating a compressed version of a given document that provides useful information to users, and multi-document summarization is to produce a summary delivering the majority of information content from a set of documents about an explicit or implicit main topic. In our study we focus on sentence based extractive document summarization. We propose the generic document summarization method which is based on sentence clustering. The proposed approach is a continue sentence-clustering based extractive summarization methods, proposed in Alguliev [Alguliev, R. M., Aliguliyev, R. M., Bagirov, A. M. (2005). Global optimization in the summarization of text documents. *Automatic Control and Computer Sciences 39*, 42–47], Aliguliyev [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)*, 18–22 December (pp. 626–629) Hong Kong, China], Alguliev and Alyguliev [Alguliev, R. M., Alyguliev, R. M. (2007). Summarization of text-based documents with a determination of latent topical sections and information-rich sentences. *Automatic Control and Computer Sciences 41*, 132–140] Aliguliyev, [Aliguliyev, R. M. (2007). Automatic document summarization by sentence extraction. *Journal of Computational Technologies 12*, 5–15.]. The purpose of present paper to show, that summarization result not only depends on optimized function, and also depends on a similarity measure. The experimental results on an open benchmark datasets from DUC01 and DUC02 show that our proposed approach can improve the performance compared to sate-of-the-art summarization approaches.

1. Introduction

The technology of automatic document summarization is maturing and may provide a solution to the information overload problem (Hahn & Mani, 2000; Mani & Maybury, 1999). Nowadays, document summarization plays an important role in information retrieval (IR). With a large volume of documents, presenting the user with a summary of each document greatly facilitates the task of finding the desired documents (Gong & Liu, 2001). Text summarization is the process of automatically creating a compressed version of a given text that provides useful information to users, and multi-document summarization is to produce a summary delivering the majority of information content from a set of documents about an explicit or implicit main topic (Wan, 2008). Authors of the paper (Radev, Hovy, & McKeown, 2002) provide the following definition for a summary:"A summary can be loosely defined as a text that is produced from one or more texts that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that. Text here is used rather loosely and can refer to speech, multimedia documents, hypertext, etc. The main goal of a summary is to present the main ideas in a document in less space. If all sentences in a text document were of equal importance, producing a summary would not be very effective, as any reduction in the size of a document would carry a proportional decrease in its informativeness. Luckily, information content in a document appears in bursts, and one can therefore distinguish between more and less informative segments. Identifying the informative segments at the expense of the rest is the main challenge in summarization". Jones (2007) assumes a tripartite processing model distinguishing three stages: source text interpretation to obtain a source representation, source representation transformation to summary representation, and summary text generation from the summary representation.

* Fax: +994 12 439 61 21.
E-mail address: a.ramiz@science.az

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.11.022

A variety of document summarization methods have been developed recently. The paper (Jones, 2007) reviews research on automatic summarizing over the last decade. This paper reviews salient notions and developments, and seeks to assess the state-of-the-art for this challenging natural language processing (NLP) task. The review shows that some useful summarizing for various purposes can already be done but also, not surprisingly, that there is a huge amount more to do.

Sentence based extractive summarization techniques are commonly used in automatic summarization to produce extractive summaries. Systems for extractive summarization are typically based on technique for sentence extraction, and attempt to identify the set of sentences that are most important for the overall understanding of a given document. In paper Salton, Singhal, Mitra, and Buckley (1997) proposed paragraph extraction from a document based on intra-document links between paragraphs. It yields a text relationship map (TRM) from intra-links, which indicate that the linked texts are semantically related. It proposes four strategies from the TRM: bushy path, depth-first path, segmented bushy path, augmented segmented bushy path. An improved version of this approach proposed in paper (Alguliev & Aliguliyev, 2005).

In our study we focus on sentence based extractive summarization. We propose the generic document summarization method which is based on sentence-clustering. The proposed approach is a continue sentence-clustering based extractive summarization methods, proposed in Alguliev, Aliguliyev, and Bagirov (2005), Aliguliyev (2006), Alguliev and Alyguliev (2007), Aliguliyev (2007). The purpose of present paper to show, that summarization result not only depends on optimized function, and also depends on a similarity measure. The experimental results on an open benchmark datasets from DUC01 and DUC02 (http://duc.nist.gov) show that our proposed approach can improve the performance compared to sate-of-the-art summarization approaches.

The rest of this paper is organized as follows: Section 2 introduces related works. The proposed sentence-clustering based approach for generic single-document summarization is presented in Section 3. The differential evolution algorithm for optimization procedure is given in Section 4. The extractive technique is represented in Section 5. The experiments and results are given in Section 6. Lastly, we conclude our paper in Section 7.

2. Related work

Generally speaking, the methods can be either extractive summarization or abstractive summarization. Extractive summarization involves assigning salience scores to some units (e.g. sentences, paragraphs) of the document and extracting the sentences with highest scores, while abstraction summarization (e.g. http://www1.cs.columbia.edu/nlp/newsblaster/) usually needs information fusion, sentence compression and reformulation (Mani & Maybury, 1999; Wan, 2008).

Sentence extraction summarization systems take as input a collection of sentences (one or more documents) and select some subset for output into a summary. This is best treated as a sentence ranking problem, which allows for varying thresholds to meet varying summary length requirements. Most commonly, such ranking approaches use some kind of similarity or centrality metric to rank sentences for inclusion in the summary – see, for example, Alguliev and Aliguliyev (2005), Alguliev et al. (2005), Aliguliyev (2006), Alguliev and Alyguliev (2007), Erkan and Radev (2004), Aliguliyev (2007), Fisher and Roark (2006), Radev, Jing, Stys, and Tam (2004), Salton, Singhal, Mitra and Buckley, 1997.

The centroid-based method (Erkan & Radev, 2004; Radev et al., 2004) is one of the most popular extractive summarization methods. MEAD (http://www.summarization.com/mead/) is an implementation of the centroid-based method for either single- or multi-document summarizing. It is based on sentence extraction. For each sentence in a cluster of related documents, MEAD computes three features and uses a linear combination of the three to determine what sentences are most salient. The three features used are centroid score, position, and overlap with first sentence (which may happen to be the title of a document). For single-documents or (given) clusters it computes centroid topic characterizations using tf–idf-type data. It ranks candidate summary sentences by combining sentence scores against centroid, text position value, and tf–idf title/lead overlap. Sentence selection is constrained by a summary length threshold, and redundant new sentences avoided by checking cosine similarity against prior ones (Zajic, Dorr, Lin, & Schwartz, 2007).

In the past, extractive summarizers have been mostly based on scoring sentences in the source document. In paper (Shen, Sun, Li, Yang, & Chen, 2007) each document is considered as a sequence of sentences and the objective of extractive summarization is to label the sentences in the sequence with 1 and 0, where a label of 1 indicates that a sentence is a summary sentence while 0 denotes a non-summary sentence. To accomplish this task, is applied conditional random field, which is a state-of-the-art sequence labeling method (Lafferty, McCallum, & Pereira, 2001). In paper Wan, Yang, and Xiao (2007) proposed a novel extractive approach based on manifold–ranking of sentences to query-based multi-document summarization. The proposed approach first employs the manifold–ranking process to compute the manifold–ranking score for each sentence that denotes the biased information-richness of the sentence, and then uses greedy algorithm to penalize the sentences with highest overall scores, which are deemed both informative and novel, and highly biased to the given query.

The summarization techniques can be classified into two groups: supervised techniques that rely on pre-existing document-summary pairs, and unsupervised techniques, based on properties and heuristics derived from the text. Supervised extractive summarization techniques treat the summarization task as a two-class classification problem at the sentence level, where the summary sentences are positive samples while the non-summary sentences are negative samples. After representing each sentence by a vector of features, the classification function can be trained in two different manners (Mihalcea & Ceylan, 2007). One is in a discriminative way with well-known algorithms such as support vector machine (SVM) (Yeh, Ke, Yang, & Meng, 2005). Many unsupervised methods have been developed for document summarization by exploiting different features and relationships of the sentences – see, for example Alguliev and Aliguliyev (2005), Alguliev et al. (2005), Aliguliyev (2006), Alguliev and Alyguliev (2007), Aliguliyev (2007), Erkan and Radev (2004), Radev et al. (2004) and the references therein.

On the other hand, summarization task can also be categorized as either generic or query-based. A query-based summary presents the information that is most relevant to the given queries (Dunlavy, O'Leary, Conroy, & Schlesinger, 2007; Fisher & Roark, 2006; Li, Sun, Kit, & Webster, 2007; Wan, 2008) while a generic summary gives an overall sense of the document's content (Alguliev & Aliguliyev, 2005; Alguliev et al., 2005; Aliguliyev, 2006; Alguliev & Alyguliev, 2007; Aliguliyev, 2007; Dunlavy et al., 2007; Gong & Liu, 2001; Jones, 2007; Li et al., 2007; Salton et al., 1997; Wan, 2008). The QCS system (Query, Cluster, and Summarize) (Dunlavy et al., 2007) performs the following tasks in response to a query: retrieves relevant documents; separates the retrieved documents into clusters by topic, and creates a summary for each cluster. QCS is a tool for document retrieval that presents results in a format so that a user can quickly identify a set of documents of interest. In paper McDonald and Chen (2006) are developed a generic, a query-based, and a hybrid summarizer, each with differing amounts of document context. The generic summarizer used a blend of discourse information and information obtained through traditional surface-level analysis. The query-based summarizer used only query-term information, and the hybrid summarizer used some discourse information along with query-term information. The article Fung and Ngai (2006) presents a multi-document, multi-lingual, theme-based summarization system based on modeling text cohesion (story flow). In this paper a Näive Bayes classifier for document summarization also proposed.

Automatic document summarization is a highly interdisciplinary research area related with computer science, multimedia, statistics, as well as cognitive psychology. In paper Guo and Stylios (2005) is introduced an intelligent system, the event-indexing and summarization (EIS) system, for automatic document summarization, which is based on a cognitive psychology model (the event-indexing model) and the roles and importance of sentences and their syntax in document understanding. The EIS system involves syntactic analysis of sentences, clustering and indexing sentences with five indices from the event-indexing model, and extracting the most prominent content by lexical analysis at phrase and clause levels.

3. Sentence clustering

Data clustering is the process of identifying natural groupings or clusters within multidimensional data based on some similarity measure. Clustering is a fundamental process in many different disciplines such as text mining, pattern recognition, IR etc. Hence, researchers from different fields are actively working on the clustering problem (Grabmeier & Rudolph, 2002; Han & Kamber, 2006; Jain, Murty, & Flynn, 1999; Omran, Engelbrecht, & Salman, 2007).

Clustering of text documents is a central problem in text mining which can be defined as grouping documents into clusters according to their topics or main contents. Document clustering has many purposes including expanding a search space, generating a summary, automatic topic extraction, browsing document collections, organizing information in digital libraries and detecting topics. A variety of approaches to document clustering have been developed. The surveys on the topics Grabmeier and Rudolph (2002), Han and Kamber (2006), Jain et al. (1999), Omran et al. (2007) offer a comprehensive summary of the different applications and algorithms.

Generally clustering problems are determined by four basic components: (1) the (physical) representation of the given data set; (2) the distance/dissimilarity measures between data points; (3) the criterion/objective function which the clustering solutions should aim to optimize; and, (4) the optimization procedure. For a given data clustering problem, the four components are tightly coupled. Various methods/criteria have been proposed over the years from various perspectives and with various focuses (Hammouda & Kamel, 2004).

3.1. Sentence representation and dissimilarity measure between sentences

Let a document D is decomposed into a set of sentences $D = (S_1, S_2,\ldots,S_n)$, where n is the number of sentences in a document D. Let $T = (t_1,t_2,\ldots,t_m)$ represent all the words (terms) occurring in a document D, where m is the number of words in a document. In most existing document clustering algorithms, documents are represented using the vector space model (VSM) (Han & Kamber, 2006), which treats a document as a bag of words. Each document is represented using these words as a vector in m-dimensional space. A major characteristic of this representation is the high-dimensionality of the feature space, which imposes a big challenge to the performance of clustering algorithms. They could not work efficiently in high-dimensional feature spaces due to the inherent sparseness of the data (Li, Luo, & Chung, 2008). If this technique were applied to sentence similarly, it should have main drawback: the sentence representation is not very efficient. The vector dimension m is very large compared to the number of words in a sentence, thus the resulting vectors would have many null components (Li et al., 2008). In our method, a sentence S_i is represented as sequence of words, $S_i = (t_1, t_2, \ldots, t_{m_i})$, instead of the bag of words, where m_i is the number of words in a sentence S_i.

Similarity measures play an increasingly important role in NLP and IR. Similarity measures have been used in text-related research and application such as text mining, information retrieving, text summarization, and text clustering. These applications show that the computation of sentence similarity has become a generic component for the research community involved in knowledge representation and discovery. In general, there is extensive literature on measuring the similarity between documents, but there are very few publications relating to the measurement of similarity between very short texts and sentences (Li et al., 2008; Liu, Zhou, & Zheng, 2007). In general, there is extensive literature on measuring the similarity between documents, but there are very few publications relating to the measure similarity between short texts or sentences (Liu et al., 2007). Liu et al. (2007) present a novel method to measure similarity between sentences by analyzing parts of speech and using Dynamic Time Warping technique. The paper Li, McLean, Bandar, O'Shea, and Crockett (2006) presents a method for measuring the similarity between sentences or very short texts, based on semantic and word order information. First, semantic similarity is derived from a lexical knowledge base and a corpus. Second, the proposed method considers the impact of word order on sentence meaning. The overall sentence similarity is defined as a combination of semantic similarity and word order similarity. In paper Wan (2007) proposed a novel measure based on the earth mover's distance (EMD) to evaluate document similarity by allowing many-to-many matching between subtopics. First, each document is decomposed into a set of subtopics, and then the EMD is employed to evaluate the similarity between two sets of subtopics for two documents by solving the transportation problem. The proposed measure is an improvement of the previous optimal matching (OM)-based measure, which allows only one-to-one matching between subtopics. In paper Bollegala, Matsuo, and Ishizuka (2007) proposed a method which integrates both page counts and snippets to measure semantic similarity between a given pair of words. In this paper modified four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice and PMI (Point-wise mutual information), to compute semantic similarity using page counts.

In this section we present a method to measure dissimilarity between sentences using the normalized google distance (NGD) (Cilibrasi & Vitányi, 2007). NGD takes advantage of the number of hits returned by Google to compute the semantic distance between concepts. The concepts are represented with their labels which are fed to the Google search engine as search terms.

First, using the NGD we define the global and local dissimilarity measure between terms (as shown in Cilibrasi & Vitányi, 2007 the NGD is nonnegative and does not satisfy the triangle inequality, i.e. hence isn't distance and consequently in the further it we shall name dissimilarity measure). According to definition NGD the global dissimilarity measure between terms t_k and t_l also is defined by the formula:

$$\mathrm{NGD}^{\mathrm{global}}(t_k, t_l) = \frac{\max\left\{\log(f_k^{\mathrm{global}}), \log(f_l^{\mathrm{global}})\right\} - \log(f_{kl}^{\mathrm{global}})}{\log N_{\mathrm{Google}} - \min\left\{\log(f_k^{\mathrm{global}}), \log(f_l^{\mathrm{global}})\right\}}, \tag{1}$$

where f_k^{global} is the number of web pages containing the search term t_k, and f_{kl}^{global} denotes the number of web pages containing both terms t_k and t_l, N_{Google} is the number of web pages indexed by Google.

The following are the main properties of the NGD (Cilibrasi & Vitányi, 2007):

(1) The range of the NGD is in *0* and ∞;
- If $t_k = t_l$ or if $t_k \neq t_l$ but frequency $f_k^{\text{global}} = f_l^{\text{global}} = f_{kl}^{\text{global}} > 0$, then $\text{NGD}^{\text{global}}(t_k,t_l) = 0$. That is, the semantics of t_k and t_l, in the Google sense is the same.
- If frequency $f_k^{\text{global}} = 0$, then for every term t_k, we have $f_{kl}^{\text{global}} = 0$, and the $\text{NGD}^{\text{global}}(t_k, t_l) = \frac{\infty}{\infty}$, which we take to be 1 by definition.
- If frequency $f_k^{\text{global}} \neq 0$ and $f_{kl}^{\text{global}} = 0$, we take $\text{NGD}^{\text{global}}(t_k,t_l) = 1$.

(2) $\text{NGD}(t_k,t_k) = 0$ for every t_k. For every pair t_k and t_l, we have $\text{NGD}^{\text{global}}(t_k,t_l) = \text{NGD}^{\text{global}}(t_l,t_k)$: It is symmetric.

Using the formula (1) we define a global dissimilarity measure between sentences S_i and S_l as follows:

$$\text{diss}_{\text{NGD}}^{\text{global}}(S_i, S_j) = \frac{\sum_{t_k \in S_i} \sum_{t_l \in S_j} \text{NGD}^{\text{global}}(t_k, t_l)}{m_i m_j}, \tag{2}$$

From the properties of NGD follows, that: (1) the range of the $\text{diss}_{\text{NGD}}^{\text{global}}(S_i, S_j)$ is in 0 and ∞; (2) If $t_k = t_l$ or if $t_k \neq t_l$ but frequency $f_k^{\text{global}} = f_l^{\text{global}} = f_{kl}^{\text{global}} > 0$, then $\text{diss}_{\text{NGD}}^{\text{global}}(S_i, S_j) = 0$; and (3) $\text{diss}_{\text{NGD}}^{\text{global}}(S_i, S_i) = 0$ for every S_i. Dissimilarity measure between sentences is exchangeable in that $\text{diss}_{\text{NGD}}^{\text{global}}(S_i, S_j) = \text{diss}_{\text{NGD}}^{\text{global}}(S_j, S_i)$ for every pair S_i and S_j.

Similarly, we define the local dissimilarity measure between sentences S_i and S_j:

$$\text{diss}_{\text{NGD}}^{\text{local}}(S_i, S_j) = \frac{\sum_{t_k \in S_i} \sum_{t_l \in S_j} \text{NGD}^{\text{local}}(t_k, t_l)}{m_i m_j}, \tag{3}$$

where

$$\text{NGD}^{\text{local}}(t_k, t_l) = \frac{\max\left\{\log(f_k^{\text{local}}), \log(f_l^{\text{local}})\right\} - \log(f_{kl}^{\text{local}})}{\log n - \min\left\{\log(f_k^{\text{local}}), \log(f_l^{\text{local}})\right\}} \tag{4}$$

is the local dissimilarity measure between terms t_k and t_l, which f_k^{local} denotes the number of sentences in a document *D*, containing the term t_k, and f_{kl}^{local} denotes the number of sentences containing both terms t_k and t_l. If the number of sentences $n = 1$, then we have $f_k^{\text{local}} = f_l^{\text{local}} = f_{kl}^{\text{local}}$ and the $\text{diss}_{\text{NGD}}^{\text{local}}(t_k, t_l) = \frac{0}{0}$, which we take to be 0 by definition.

Thus, the overall sentence dissimilarity is defined as a product of global and local dissimilarity measures:

$$\text{diss}_{\text{NGD}}(S_i, S_j) = \text{diss}_{\text{NGD}}^{\text{local}}(S_i, S_j) \cdot \text{diss}_{\text{NGD}}^{\text{local}}(S_i, S_j). \tag{5}$$

3.2. Criterion function

Typically clustering algorithms can be categorized as agglomerative or partitional based on the underlying methodology of the algorithm, or as hierarchical or flat (non-hierarchical) based on the structure of the final solution (Grabmeier & Rudolph, 2002; Han & Kamber, 2006; Jain et al., 1999; Omran et al., 2007). A key characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process. In recent years, it has been recognized that the partitional clustering technique is well suited for clustering a large document database due to their relatively low computational requirements (Zhao & Karypis, 2004).

Automatic clustering is a process of dividing a set of objects into unknown groups, where the best number *k* of groups (or clusters) is determined by the clustering algorithm. That is, objects within each group should be highly similar to each other than to objects in any other group. The automatic clustering problem can be defined as follows (Das, Abraham, & Konar, 2008; Grabmeier & Rudolph, 2002; Han & Kamber, 2006; Jain et al., 1999; Omran et al., 2007):

The set of sentences $D = (S_1, S_2, \ldots, S_n)$ are clustered into non-overlapping groups $C = \{C_1, \ldots, C_k\}$, where *C* is called a cluster, *k* is the unknown number of clusters. The partition should maintain three properties:

(1) Two different clusters should have no sentences in common, i.e. $C_p \cap C_q = \emptyset$ for $\forall p \neq q \quad p,q \in \{1, 2, \ldots, k\}$;
(2) Each sentence should definitely be attached to a cluster, i.e. $\bigcup_{p=1}^{k} C_p = D$;
(3) Each cluster should have at least one sentence assigned, i.e. $C_p \neq \emptyset \quad \forall p \in \{1, 2, \ldots, k\}$.Partitional clustering can be viewed as an optimization procedure that tries to create high-quality clusters according to a particular criterion function. Criterion functions used in partitional clustering reflect the underlying definition of the "goodness" of clusters. Many criterion functions have been proposed in the literature (Grabmeier & Rudolph, 2002; Han & Kamber, 2006; Jain et al., 1999; Omran et al., 2007; Zhao & Karypis, 2004) to produce more balanced partitions.

We introduce a criterion function that is defined as follows:

$$F = \frac{\sum_{p=1}^{k} \mid C_p \mid \sum_{S_i, S_j \in C_p} \text{diss}_{\text{NGD}}(S_i, S_j)}{\sum_{p=1}^{k-1} \sum_{q=p+1}^{k} \sum_{S_i \in C_p} \sum_{S_j \in C_q} \mid C_p \mid\mid C_q \mid \text{diss}_{\text{NGD}}(S_i, S_j)} \to \min, \tag{6}$$

where $|C_p|$ is the number of sentences assigned to cluster C_p.

The criterion function (6) optimizes both intra–cluster similarity and inter–cluster dissimilarity. This function is obtained by combining two criterions:

$$F_1 = \sum_{p=1}^{k} \mid C_p \mid \sum_{S_i, S_j \in C_p} \text{diss}_{\text{NGD}}(S_i, S_j) \to \min, \tag{7}$$

and

$$F_2 = \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} \sum_{S_i \in C_p} \sum_{S_j \in C_q} \mid C_p \mid\mid C_q \mid \text{diss}_{\text{NGD}}(S_i, S_j) \to \max. \tag{8}$$

The F_1 criterion function (7) minimizes the sum of the average pairwise similarity between the sentences assigned to each cluster. The F_2 criterion function (8) computes the clustering by finding a solution that separates each cluster from other cluster. Specifically, it tries to maximize the dissimilarity between the sentences S_i and S_j assigned to different clusters C_p and C_q, $(p \neq q)$, respectively. In these criterion functions each cluster weighted according to its cardinality.

3.3. Estimating the number of clusters

Determination of the optimal number of clusters in a data set is a difficult issue and depends on the adopted validation and chosen similarity measure, as well as on data representation. For clustering of sentences, customers can't predict the latent topic number in the document, so it's impossible to offer *k* effectively. The strategy that we used to determine the optimal number of clusters (the number of topics in a document) is based on the distribution of words in the sentences:

$$k = n \frac{\mid D \mid}{\sum_{i=1}^{n} \mid S_i \mid} = n \frac{\mid \bigcup_{i=1}^{n} S_i \mid}{\sum_{i=1}^{n} \mid S_i \mid}, \tag{9}$$

where $|A|$ is the number of terms in the document (sentence) A.

In words, the number of clusters (i.e. the number of topics in a document) is defined as n times the ratio of the total number of terms in the document to the cumulative number of terms in the sentences considered separately.

Let us analyze the properties of this estimation by examining some particular cases:

Document of identical sentences. The document is constituted by n sentences having the same set of terms. Therefore, the set of terms of the document coincides with the set of terms of each sentence: $D = (t_1,t_2,\ldots,t_m) = S_i = S$. From the definition (9) follows that $k = n\frac{|\bigcup_{i=1}^{n} S_i|}{\sum_{i=1}^{n}|S_i|} = n\frac{|\bigcup_{i=1}^{n} S|}{\sum_{i=1}^{n}|S|} = n\frac{|S|}{\sum_{i=1}^{n}|S|} = 1$. An intuitively appealing result, since the document corresponds actually to a collection of single sentences.

Note that the converse of this property is also true, that is, if the number of topics (k) of a document is unitary, all the sentences have necessarily the same terms. This can be proved by the following argument. If $k = 1$ the from the definition (9) follows that $n \mid D \mid = n \mid \bigcup_{i=1}^{n} S_i \mid = n \mid \bigcup_{i=1}^{n} S \mid = \sum_{i=1}^{n} \mid S \mid$.

Document of pairwise maximally distinct sentences. The document is constituted by sentences, do not have any term in common, that is, $S_i \cap S_j = \emptyset$ for $i \neq j$. This means that each term belonging to $D = \bigcup_{i=1}^{n} S_i$ belongs only to one of the sentences S_i and therefore $\mid D \mid = \mid \bigcup_{i=1}^{n} S_i \mid = \sum_{i=1}^{n} \mid S_i \mid$, from which follows that $k = n$.

As before, the converse is also true, that is, if $k = n$ the sentences have, pairwise, no terms in common. This can be proved by the following argument. If $k = n$then from the definition (9) follows that $\mid D \mid = \mid \bigcup_{i=1}^{n} S_i \mid = \sum_{i=1}^{n} \mid S_i \mid$. Let us assume that there exists a pair of sentences such that $S_i \cap S_j \neq \emptyset$. This means that there exists at least one term that belongs to both sentences. This term will be counted only once in $\mid \bigcup_{i=1}^{n} S_i \mid$, but at least twice in $\sum_{i=1}^{n} \mid S_i \mid$. Thus the condition $\mid \bigcup_{i=1}^{n} S_i \mid = \sum_{i=1}^{n} \mid S_i \mid$ could not be realized, which contracts our assumption.

With analogues deductions, it can be proved that the values of the number of clusters obtained in these two cases constitute actually a bound for k, that is, that we always have $1 \leqslant k \leqslant n$. This fact, along with the interpretation of formula (9) in terms of average number of terms that will be presented shortly, suggests the interpretation of the value of k as the number of equivalent sentences of the document.

4. A discrete differential evolution for clustering

There are many techniques that can be used to optimize the criterion functions (6)–(8) described in the previous Section 3. In our study these criterion functions were optimized using a differential evolution (Das et al., 2008; Storn & Price, 1997). The execution of the differential evolution is similar to other evolutionary algorithms like genetic algorithms or evolution strategies. The evolutionary algorithms differ mainly in the representation of parameters (usually binary strings are used for genetic algorithms while parameters are real-valued for evolution strategies and differential evolution) and in the evolutionary operators.

4.1. The basic differential evolution algorithm

The classical DE (Storn & Price, 1997) is a population-based global optimization that uses a real-coded representation. Like to other evolutionary algorithm, DE also starts with a population of Pop n-dimensional search variable vectors. The rth individual vector (chromosome) of the population at time-step (generation) t has n components, i.e.,

$$X_r(t) = [x_{r,1}(t), x_{r,2}(t), \ldots, x_{r,n}(t)], \quad r = 1, 2, \ldots, \text{Pop}. \tag{10}$$

For each individual vector $X_r(t)$ that belongs to the current population, DE randomly samples three other individuals, i.e., $X_{r1}(t)$, $X_{r2}(t)$, and $X_{r3}(t)$ from the same generation (for mutually different $r \neq r1 \neq r2 \neq r3$). It then calculates the (componentwise) difference of $X_{r2}(t)$ and $X_{r3}(t)$, scales it by a scalar λ, and creates a trial offspring $Y_r(t+1) = (y_{r,1}(t+1), y_{r,2}(t+1), \ldots, y_{r,d}(t+1))$ by adding the result to $X_{r1}(t)$. Thus, for the sth component of each vector

$$y_{r,s}(t+1) = \begin{cases} x_{r1,s}(t) + \lambda(x_{r2,s}(t) - x_{r3,s}(t)), & \text{if } \text{rnd}_s < CR \\ x_{r,s}(t), & \text{otherwise} \end{cases}. \tag{11}$$

The scaling factor (λ) and the crossover rate (CR) are control parameters of DE, are set by the user. Both values remain constant during the search process. λ is a real-valued factor (usually in range [0,1]), that controls the amplification of differential variations and CR is a real-valued crossover factor in range [0,1] controlling the probability to choose mutated value for x instead of its current value. rnd_s is the uniformly distributed random numbers within the range [0,1] chosen once for each $s \in \{1,2,\ldots,n\}$.

If the new offspring yields a better value of the objective function, it replaces its parent in the next generation; otherwise, the parent is retained in the population, i.e.,

$$X_r(t+1) = \begin{cases} Y_r(t+1), & \text{if } \boldsymbol{F}(Y_r(t+1)) < \boldsymbol{F}(X_r(t)) \\ X_r(t), & \text{if } \boldsymbol{F}(Y_r(t+1)) \geqslant \boldsymbol{F}(X_r(t)) \end{cases}, \tag{12}$$

where $\boldsymbol{F}(\cdot)$ is the objective function to be minimized.

4.2. Chromosome encoding

We use a genetic encoding that directly allocates n objects to k clusters, such that each candidate solution consists of n genes, each with an integer value in the range $[1,k]$. For example, for $n = 7$ and $k = 3$, the encoding [2,3,3,2,1,2,1,1,3,2] allocates the fifth, seventh and eighth objects to cluster 1, the first, fourth, sixth and tenth objects to cluster 2, and the second, third and ninth objects to cluster 3.

For representing the a th chromosome of the population at the current generation (at time t) here the following notation has been used:

$$X_a(t) = [x_{a,1}(t), x_{a,2}(t), \ldots, x_{a,n}(t)], \tag{13}$$

where $x_{r,s}(t) \in \{1,2,\ldots,k\}$ is an integer number, $a = 1,\ldots,\text{Pop}$, Pop is the size of the population.

4.3. Fitness computation

To judge the quality of a partition provided by a chromosome, it is necessary to have a fitness functions. The fitness functions are defined as

$$\text{fitness}_1(X_a) = \frac{1}{F_1(X_a)}, \tag{14}$$

$$\text{fitness}_2(X_a) = F_2(X_a), \tag{15}$$

$$\text{fitness}(X_a) = \frac{1}{F(X_a)}, \tag{16}$$

so that maximization of the fitness functions (14)–(16) leads to minimization (or maximization) of the criterion functions (6)–(8).

In its classical form, the DE algorithm is only applicable for optimization of continues variables. In our study we adapt it for optimization of discrete variables.

4.4. Population initialization

A natural way to initialize the initial population (at time $t = 0$) is to seed it within random values within the given range $(1,k+1)$, e.g.,

$$X_a(0) = [x_{a,1}(0), x_{a,2}(0), \dots, x_{a,n}(0)],$$
$$x_{a,r}(0) = k_r \cdot \mathrm{sigm}(k_r) + 1, \quad a = 1, 2, \dots, P, \quad r = 1, 2, \dots, n, \tag{17}$$

where k_r is the uniformly distributed random value within range [1,k] chosen once for each r = 1,2,. . .,n and *sigm*(z) is a sigmoid function that maps from the real numbers into [0,1]. It has the properties that $sigm(0) = \frac{1}{2}$ and $sigm(z) \to 1$ as $z \to \infty$. It is mathematically formulated as,

$$\mathrm{sigm}(z) = \frac{1}{1 + \exp(-z)}. \tag{18}$$

4.5. Crossover

We propose a modified version of the classical differential evolution. The proposed version for the chromosome of the current best solution $X_b(t)$, randomly chooses two other chromosomes $X_u(t)$ and $X_v(t)$ (b,u,v ∈ {1,2,. . .,P} and $b \neq u \neq v$) from the same generation. It then calculates the weighted difference $\pi_c X_u(t) - (1 - \pi_c)\ X_v(t)$ and creates a trial offspring chromosome by adding the result to the chromosome of $X_b(t)$. Thus for the sth gene $y_{b,s}(t+1)$of child chromosome $Y_b(t+1)$, we have

$$y_{b,s}(t+1) = \begin{cases} x_{b,s}(t) + \pi_s x_{u,s}(t) - (1 - \pi_s) x_{v,s}(t), & \text{if } \mathrm{rnd}_s < CR \\ x_{b,s}(t), & \text{otherwise} \end{cases} \tag{19}$$

where rnd_s and π_s are a uniformly distributed random numbers within the range [0,1] chosen once for each s = 1,2,. . .,n.

Real value of the s th gene it will be converted to integer value as follows:

$$y_{b,s}(t+1) = \begin{cases} \mathrm{INT}(k \cdot \mathrm{rnd}_s + 1), & \text{if } \mathrm{INT}(y_{b,s}(t+1)) < 1 \\ \quad \text{or } \mathrm{INT}(y_{b,s}(t+1)) > k \\ \mathrm{INT}(y_{b,s}(t+1)), & \text{otherwise} \end{cases}, \tag{20}$$

where INT(·) is a function for converting a real value to an integer value by truncation.

To keep the population size constant over subsequent generations, the next step of the algorithm calls for selection to determine which one between the parent and child will survive in the next generation (at time $t+1$). Differential evolution uses the principle of "survival of the fittest" in its selection process which may be expressed as:

$$X_b(t+1) = \begin{cases} Y_b(t+1), & \text{if } \mathrm{fitness}(Y_b(t+1)) > \mathrm{fitness}(X_b(t)) \\ X_b(t), & \text{otherwise} \end{cases}. \tag{21}$$

If the new offspring yields a better value of the fitness function, it replaces its parent in the next generation; otherwise the parent is retained in the population.

4.6. Mutation

For target chromosome (i.e. the chromosome of the best solution $X_b\ (t)$) a mutant chromosome is generated according to

$$y_{b,q}(t+1) = \begin{cases} x_{b,q}(t) + \pi_q x_{b,r}(t) - (1 - \pi_q) x_{b,s}(t), & \text{if } \mathrm{rnd}_q < MR \\ x_{b,q}(t), & \text{otherwise} \end{cases}, \tag{22}$$

with random indexes q,r,s ∈ {1,2,. . .,n}, integer, mutually different, $q \neq r \neq s$. $MR \in [0,1]$ is the predefined mutation rate, rnd_q and π_q are a uniformly distributed random numbers within the range [0,1] chosen once for each q = 1,2,. . .,n.

Similarly to (20) real value of the q th gene it will be converted to integer value

$$y_{b,q}(t+1) = \begin{cases} \mathrm{INT}(k * \mathrm{rnd}_q + 1), & \text{if } \mathrm{INT}(y_{b,q}(t+1)) < 1 \\ \quad \text{or } \mathrm{INT}(y_{b,q}(t+1)) > k \\ \mathrm{INT}(y_{b,q}(t+1)), & \text{otherwise} \end{cases}. \tag{23}$$

If the mutant chromosome yields a better value of the fitness function, it replaces its parent in the next generation; otherwise the parent is retained in the population.

4.7. Termination criterion

The termination criterion of differential evolution could be a given number of consecutive iterations within which no improvement on solutions, a specified CPU time limit, or maximum number of iterations (fitness calculation), $t_{\max}$, is attained. Unless otherwise specified, in this paper we use the last one as the termination criteria, i.e. the algorithm terminates when a maximum number of fitness calculation is achieved.

5. A sentence extract technique

Extractive summarization works by choosing a subset of the sentences in the original document. This process can be viewed as identifying the most salient sentences in a cluster that give the necessary and sufficient amount of information related to main content of the cluster. In a cluster of related sentences, many of the sentences are expected to be somewhat similar to each other since they are all about the same topic. The approach, proposed in papers Erkan and Radev (2004), Radev et al. (2004), is to assess the centrality of each sentence in a cluster and extract the most important ones to include in the summary. In centroid-based summarization, the sentences that contain more words from the centroid of the cluster are considered as central. Centrality of a sentence is often defined in terms of the centrality of the words that it contains. In this section we use other criterion to assess sentence salience, proposed in paper (Pavan & Pelillo, 2007).

Let C_p be nonempty cluster and $S_i \in C_p$. Then the average weighted degree of S_i with respect to cluster C_p is defined as

$$\mathrm{awdeg}_{C_p}(S_i) = \frac{1}{\mid C_p \mid} \sum_{S_j \in C_p} \mathrm{diss}_{\mathrm{NGD}}(S_i, S_j). \tag{24}$$

Observe that $\mathrm{awdeg}_{(S_i)}(S_i) = 0$ for any $S_i \in C_p$. Moreover, if $S_j \notin C_p$, we define:

$$\Phi_{C_p}(S_i, S_j) = \mathrm{diss}_{\mathrm{NGD}}(S_i, S_j) - \mathrm{awdeg}_{C_p}(S_i). \tag{25}$$

From $\mathrm{awdeg}_{(S_i)}(S_i) = 0$ follows that $\Phi_{S_i}(S_i, S_j) = \mathrm{diss}_{\mathrm{NGD}}(S_i, S_j)$ for all S_i,S_j ∈ C_p, with $i \neq\ j$. Intuitively, $\Phi_{C_p}(S_i, S_j)$ measures the relative measure between sentences S_j and S_i, with respect to the average measure between S_i and its neighbors in cluster C_p. Note that $\Phi_{C_p}(S_i, S_j)$ can be either positive or negative.

Thus the weight of sentence S_i with respect to cluster C_p will be defined by the following recursive formula as

$$W_{C_p}(S_i) = \begin{cases} 1, & \text{if } \mid C_p \mid = 1 \\ \sum\limits_{S_j \in C_p \backslash \{S_i\}} \Phi_{C_p \backslash \{S_i\}}(S_j, S_i) W_{C_p \backslash \{S_i\}}(S_j), & \text{otherwise} \end{cases}. \tag{26}$$

Note that $W_{\{S_i,S_j\}}(S_i) = W_{\{S_i,S_j\}}(S_j) = \mathrm{diss}_{\mathrm{NGD}}(S_i, S_j)$ for all S_i,S_j ∈ C_p($i \neq j$). Intuitively, $W_{C_p}(S_i)$ gives us a measure of the overall (relative) dissimilarity measure between sentence S_i and the sentences of $C_p \backslash \{S_i\}$ with respect to the overall measure among the sentences in $C_p \backslash \{S_i\}$.

Finally, as to selection of sentences to generate a summary, in each cluster sentences are ranked in reversed order of their score and the top ranked sentences are selected for in the extractive summary.

6. Experiments and results

In this section, we conduct experiments to test our summarization method empirically.

6.1. Datasets

For evaluation the performance of our methods we used two document datasets DUC01 and DUC02 and corresponding 100-word summaries generated for each of documents. The DUC01 and DUC02 are an open benchmark datasets which contain 147 and 567 documents-summary pairs from Document Understanding Conference (http://duc.nist.gov). We use them because they are for generic single-document extraction that we are interested in and they are well preprocessed. These datasets DUC01 and DUC02 are clustered into 30 and 59 topics, respectively. In those document datasets, stopwords were removed using the stoplist provided in ftp://ftp.cs.cornell.edu/pub/smart/english.stop and the terms were stemmed using Porter's scheme (Porter, 1980), which is a commonly used algorithm for word stemming in English.

6.2. Evaluation metrics

There are many measures that can calculate the topical similarities between two summaries. For evaluation the results we use two methods. The first one is by *precision* (P), *recall* (R) and F_1-measure which are widely used in Information Retrieval. For each document, the manually extracted sentences are considered as the reference summary (denoted by Summ_{ref}). This approach compares the candidate summary (denoted by Summ_{cand}) with the reference summary and computes the P, R and F_1-measure values as shown in formula (27) (Shen et al., 2007)

$$P = \frac{|\,\text{Summ}_{\text{ref}} \cap \text{Summ}_{\text{cand}}\,|}{|\,\text{Summ}_{\text{cand}}\,|},$$
$$R = \frac{|\,\text{Summ}_{\text{ref}} \cap \text{Summ}_{\text{cand}}\,|}{|\,\text{Summ}_{\text{ref}}\,|}, \quad F_1 = \frac{2PR}{P+R}. \tag{27}$$

The second measure we use the ROUGE toolkit (Lin et al., 2003; Lin, 2004) for evaluation, which was adopted by DUC for automatically summarization evaluation. It has been shown that ROUGE is very effective for measuring document summarization. It measures summary quality by counting overlapping units such as the N-gram, word sequences and word pairs between the candidate summary and the reference summary. The ROUGE-N measure compares N-grams of two summaries, and counts the number of matches. The measure is defined by formula (28) (Lin et al., 2003; Lin, 2004; Nanba & Okumura, 2006; Svore, Vanderwende, & Burges, 2007)

$$\text{ROUGE}-\text{N} = \frac{\sum_{S \in \text{Summ}_{\text{ref}}} \sum_{\text{N}-\text{gram} \in S} \text{Count}_{\text{match}}(\text{N}-\text{gram})}{\sum_{S \in \text{Summ}_{\text{ref}}} \sum_{\text{N}-\text{gram} \in S} \text{Count}(\text{N}-\text{gram})}, \tag{28}$$

where N stands for the length of the N-gram, $\text{Count}_{match}(\text{N}-\text{gram})$ is the maximum number of N-grams co-occurring in candidate summary and a set of reference–summaries. $\text{Count}(\text{N}-\text{gram})$ is the number of N-grams in the reference summaries. We use two of the ROUGE metrics in the experimental results, ROUGE-1 (unigram-based) and ROUGE-2 (bigram-based).

6.3. Simulation strategy and parameters

The optimization procedure used here is stochastic in nature. Hence, for each criterion function (F_1, F_2 and F) it has been run several times. The parameters of the DE are set as follows: the population size, Pop = 200; the number of iteration (fitness evaluation), $t_{\max} = 1000$; the crossover rate, $CR = 0.6$; the mutation rate, $MR = 0.2$. The results reported in this section are averages over 20 runs for each criterion functions. Finally, we would like to point out that algorithm was developed from scratch in Delphi 7 platform on a Pentium Dual CPU, 1.6 GHz PC, with 512 KB cache, and 1 GB of main memory in Windows XP environment.

6.4. Performance evaluation and discussion

The first experiment compares our criterion functions F_1, F_2 and F with four methods CRF (Shen et al., 2007), NetSum (Svore et al., 2007), Manifold–Ranking (Wan et al., 2007) and SVM (Yeh et al., 2005). Tables 1 and 2 show the results of all the methods in terms ROUGE-1, ROUGE-2, and F_1-measure metrics on DUC01 and DUC02 datasets, respectively. As shown in Tables 1 and 2, on DUC01 dataset, the average values of ROUGE-1, ROUGE-2 and F_1 metrics of all the methods are better than on DUC02 dataset. As seen from Tables 1 and 2 Manifold–Ranking is the worst method, while our criterion function F is the best of both evaluation metrics. In the Tables 1 and 2 highlighted (***bold italic***) entries represent the best performing methods in terms of average evaluation metrics. The criterion functions F_1 and F_2, and the methods NetSum and CRF show almost identical results. Among the methods NetSum, CRF, SVM and Manifold–Ranking the best result shows NetSum.

Comparison our methods with four methods CRF, NetSum, Manifold–Ranking and SVM are shown in Tables 3 and 4. Here we use relative improvement $\frac{(\text{our method}-\text{other methods})}{\text{other methods}} \times 100$ for comparison. In the Tables 3 and 4 "+" means the result outperforms and "-" means the opposite. In spite of the fact that among our criterion functions the worst result is obtained by criterion function F_2, but it shows better result than the methods CRF, SVM and Manifold–Ranking. The criterion function F_2 concedes only to the method NetSum. Compared with the best method NetSum, on DUC01 (DUC02) dataset the criterion function F improves the performance by 3.08% (3.85%), 4.70% (10.75%) and 2.24% (3.61%) in terms ROUGE-1, ROUGE-2 and F_1, respectively.

The second experiment is to test the effectiveness of the NGD-based dissimilarity measure. We compare the results of our methods using different dissimilarity measure in particular Euclidean distance and NGD-based measure. Results of experiments are

Table 1
Average values of evaluation metrics for summarization methods (DUC01 dataset).

Methods	Average ROUGE-1	Average ROUGE-2	Average F_1-measure
F	***0.47856***	***0.18528***	***0.48324***
F_1	0.46652	0.17731	0.47635
F_2	0.46231	0.17672	0.46957
NetSum	0.46427	0.17697	0.47267
CRF	0.45512	0.17327	0.46435
SVM	0.44628	0.17018	0.45357
Manifold–Ranking	0.43359	0.16635	0.44368

Table 2
Average values of evaluation metrics for summarization methods (DUC02 dataset).

Methods	Average ROUGE-1	Average ROUGE-2	Average F_1-measure
F	***0.46694***	***0.12368***	***0.47947***
F_1	0.45658	0.11364	0.46931
F_2	0.44289	0.11065	0.46097
NetSum	0.44963	0.11167	0.46278
CRF	0.44006	0.10924	0.46046
SVM	0.43235	0.10867	0.43095
Manifold–Ranking	0.42325	0.10677	0.41657

Table 3
Performance evaluation compared between our methods and other methods (DUC01 dataset).

Methods	Metrics	NetSum	CRF	SVM	Manifold–Ranking
F	ROUGE-1	3.08 (+)	5.15 (+)	7.23 (+)	10.37 (+)
	ROUGE-2	4.70 (+)	6.93 (+)	8.87 (+)	11.38 (+)
	F_1-measure	2.24 (+)	4.07 (+)	6.54 (+)	8.92 (+)
F_1	ROUGE-1	0.48 (+)	2.50 (+)	4.54 (+)	7.59 (+)
	ROUGE-2	0.19 (+)	2.33 (+)	4.19 (+)	6.59 (+)
	F_1-measure	0.78 (+)	2.58 (+)	5.02 (+)	7.36 (+)
F_2	ROUGE-1	0.42 (-)	1.58 (+)	3.59 (+)	6.62 (+)
	ROUGE-2	0.14 (-)	1.99 (+)	3.84 (+)	6.23 (+)
	F_1-measure	0.66 (-)	1.12 (+)	3.53 (+)	5.84 (+)

Table 4
Performance evaluation compared between our methods and other methods (DUC02 dataset).

Methods	Metrics	NetSum	CRF	SVM	Manifold–Ranking
F	ROUGE-1	3.85 (+)	6.11 (+)	8.00 (+)	10.32 (+)
	ROUGE-2	10.75 (+)	13.22 (+)	13.81 (+)	15.84 (+)
	F_1-measure	3.61 (+)	4.13 (+)	11.26 (+)	15.10 (+)
F_1	ROUGE-1	1.55 (+)	3.75 (+)	5.60 (+)	7.87 (+)
	ROUGE-2	1.76 (+)	4.03 (+)	4.57 (+)	6.43 (+)
	F_1-measure	1.41 (+)	1.92 (+)	8.90 (+)	12.66 (+)
F_2	ROUGE-1	1.50 (-)	0.64 (+)	2.44 (+)	4.64 (+)
	ROUGE-2	0.91 (-)	1.29 (+)	1.82 (+)	3.63 (+)
	F_1-measure	0.39 (-)	0.11 (+)	6.97 (+)	10.66 (+)

Table 5
Comparison of evaluation metrics values for NGD-based measure and Euclidean distance (DUC01 dataset).

Methods	Dissimilarity measure	Average ROUGE-1	Average ROUGE-2	Average F_1-measure
F_1	NGD-based	***0.46652*** (+***5.15***%)	***0.17731*** (+***8.33***%)	***0.47635*** (+***2.11***%)
	Euclidean	0.44367	0.16367	0.46647
F_2	NGD-based	***0.46231*** (+***2.89***%)	***0.17672*** (+%***6.80***)	***0.46957*** (+***3.60***%)
	Euclidean	0.44934	0.16547	0.45324
F	NGD-based	***0.47856*** (+***3.26***%)	***0.18528*** (+***9.07***%)	***0.48324*** (+***1.73***%)
	Euclidean	0.46347	0.16987	0.47504

Table 6
Comparison of evaluation metrics values for NGD-based measure and Euclidean distance (DUC02 dataset).

Methods	Dissimilarity measure	Average ROUGE-1	Average ROUGE-2	Average F_1-measure
F_1	NGD-based	***0.45658*** (+***4.65***%)	***0.11364*** (+***6.72***%)	***0.46931*** (+***3.39***%)
	Euclidean	0.43628	0.10648	0.45394
F_2	NGD-based	***0.44289*** (+***5.03***%)	***0.11065*** (+***7.46***%)	***0.46097*** (+***4.00***%)
	Euclidean	0.42167	0.10297	0.44324
F	NGD-based	***0.46694*** (+***3.75***%)	***0.12368*** (+***5.48***%)	***0.47947*** (+***2.58***%)
	Euclidean	0.45007	0.11725	0.46741

reported in Tables 5 and 6. As seen from these Tables the NGD-based measure outperforms the Euclidean distance. In Tables 5 and 6 the numbers in brackets specify percent of improvement of results.

7. Conclusion

We have presented the approach to automatic document summarization based on clustering and extraction of sentences. Our approach consists of two steps. First sentences are clustered, and then on each cluster representative sentences are defined. In our study we developed a discrete differential evolution algorithm to optimize the objective functions. When comparing our methods with several existing summarization methods on an open DUC01 and DUC01 datasets, we found that our methods can improve the summarization results significantly. The methods were evaluated using ROUGE-1, ROUGE-2 and F_1 metrics. In this paper we also demonstrated that the summarization result depends on the similarity measure. Results of experiment have showed that proposed by us NGD-based dissimilarity measure outperforms the Euclidean distance.

References

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

Alguliev, R. M., & Alyguliev, R. M. (2007). Summarization of text-based documents with a determination of latent topical sections and information-rich sentences. *Automatic Control and Computer Sciences, 41*, 132–140.

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)*, 19–22 September (pp. 264–271), France.

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)*, 18–22 December (pp. 626–629), Hong Kong, China.

Aliguliyev, R. M. (2007). Automatic document summarization by sentence extraction. *Journal of Computational Technologies, 12*, 5–15.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proceedings of 16th world wide web conference (WWW16)*, May 8–12 (pp. 757–766) Banff, Alberta, Canada.

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

Das, S., Abraham, A., & Konar, A. (2008). Automatic clustering using an improved differential evolution algorithm. *IEEE Transaction on Systems, Man, and Cybernetics – Part A: Systems and Humans, 38*, 218–237.

Dunlavy, D. M., O'Leary, D. P., Conroy, J. M., & Schlesinger, J. D. (2007). QCS: A system for querying, clustering and summarizing documents. *Information Processing and Management, 43*, 1588–1605.

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

Fisher, S., & Roark, B. (2006). Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proceedings of the document understanding workshop (DUC 2006)*, 8–9 June (pp. 8) New York, USA.

Fung, P., & Ngai, G. (2006). One story, one flow: Hidden Markov story models for multilingual multidocument summarization. *ACM Transaction on Speech and Language Processing, 3*, 1–16.

Gong, Y., & Liu, X. (2001). Creating generic text summaries. In *Proceedings of the 6th international conference on document analysis and recognition (ICDAR'01)*, 10–13 September (pp. 903–907) Seattle, USA.

Grabmeier, J., & Rudolph, A. (2002). Techniques of cluster algorithms in data mining. *Data Mining and Knowledge Discovery, 6*, 303–360.

Guo, Y., & Stylios, G. (2005). An intelligent summarization system based on cognitive psychology. *Information Sciences, 174*, 1–36.

Hahn, U., & Mani, I. (2000). The challenges of automatic summarization. *IEEE Computer, 33*, 29–36.

Hammouda, K. M., & Kamel, M. S. (2004). Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering, 16*, 1279–1296.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and technique* (2nd ed.). San Francisco: Morgan Kaufman.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*, 264–323.

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

Lafferty, J. D., McCallum, & A., Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th international conference on machine learning*, 28 June–01 July (pp. 282–289).

Li, J., Sun, L., Kit, C., & Webster, J. (2007). A query-focused multi-document summarizer based on lexical chains. In *Proceedings of the document understanding conference 2007 (DUC 2007)*, 26–27 April (p. 4.) New York, USA.

Li, Y., Luo, C., & Chung, S. M. (2008). Text clustering with feature selection by using statistical data. *IEEE Transactions on Knowledge and Data Engineering, 20*, 641–652.

Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering, 18*, 1138–1150.

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

Lin, C. -Y., & Hovy, E. H. (2003). Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology (HLT-NAACL 2003)*, 27 May–1 June (Vol. 1. pp. 71–78.) Edmonton, Canada.

Liu, X., Zhou, & Y., Zheng, R. (2007). Sentence similarity based on dynamic time warping. In *Proceedings of the first international conference on semantic computing (ICSC 2007)*, 17–19 September (pp. 250–256) Irvine, USA.

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

McDonald, D. M., & Chen, H. (2006). Summary in context: Searching versus browsing. *ACM Transactions on Information Systems, 24*, 111–141.

Mihalcea, R., & Ceylan, H. (2007). Explorations in automatic book summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007)*, 28–30 June (pp. 380–389) Prague, Czech Republic.

Nanba, H., & Okumura, M. (2006). An automatic method for summary evaluation using multiple evaluation results by a manual method. In *Proceedings of the COLING/ACL on main conference poster sessions*, 17–18 July (pp. 603–610) Sydney, Australia.

Omran, M. G. H., Engelbrecht, A. P., & Salman, A. (2007). An overview of clustering methods. *Intelligent Data Analysis, 11*, 583–605.

Pavan, M., & Pelillo, M. (2007). Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Learning, 29*, 167–172.

Porter, M. (1980). An algorithm for suffix stripping. *Program, 14*, 130–137.

Radev, D., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*, 399–408.

Radev, D. R., Jing, H., Stys, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management, 40*, 919–938.

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

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

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359.

Svore, K. M., Vanderwende, L., & Burges, C. J. C. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007)*, 28–30 June (pp. 448–457) Prague, Czech Republic.

Wan, X. (2007). A novel document similarity measure based on earth mover's distance. *Information Sciences, 177*, 3718–3730.

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

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)*, January 6–12 (pp. 2903–2908) Hyderabad, India.

Yeh, J-Y., Ke, H-R., Yang, W-P., & Meng, I-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management, 41*, 75–95.

Zajic, D., Dorr, B. J., Lin, J., & Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management, 43*, 1549–1570.

Zhao, Y., & Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning, 55*, 311–331.