Noname manuscript No.
(will be inserted by the editor)

Multi-document summarization via Archetypal Analysis of the content-graph joint model

Ercan Canhasi · Igor Kononenko

Received: Dec 15, 2012 / Revised: Jun 03, 2013 / Accepted: Sep 01, 2013

Abstract In recent years, algebraic methods, more precisely matrix decomposition approaches have become a key tool for tackling document summarization problem. Typical algebraic methods used in multi-document summarization (MDS) vary from soft and hard clustering approaches to low rank approximations. In this paper, we present a novel summarization method AASum which employs the Archetypal Analysis for generic multi-document summarization. Archetypal Analysis (AA) is a promising unsupervised learning tool able to completely assemble the advantages of clustering and the flexibility of matrix factorization. In document summarization, given a content-graph data matrix representation of a set of documents, positively and/or negatively salient sentences are values on the data set boundary. These extreme values, archetypes, can be computed using AA. While each sentence in a data set is estimated as a mixture of archetypal sentences, the archetypes themselves are restricted to being sparse mixtures, i.e. convex combinations of the original sentences. Since AA in this way readily offers soft clustering, we suggest to consider it as a method for simultaneous sentence clustering and ranking. Another important argument in favor of using AA in MDS is that in contrast to other factorization methods which extract prototypical, characteristic, even basic sentences, AA selects distinct (archetypal) sentences, thus induces variability and diversity in produced summaries. Experimental results on the DUC generic summarization datasets evidence the improvement of the proposed approach over the other closely related methods.

Keywords Document summarization · Archetypal Analysis · Matrix decomposition · Content-graph joint model

E. Canhasi
Faculty of Computer and Information Science, University of Ljubljana, Slovenia;
E-mail: ercan.canhasi@uni-prizren.com

I. Kononenko
Faculty of Computer and Information Science, University of Ljubljana, Slovenia;
E-mail: igor.kononenko@fri.uni-lj.si

1 Introduction

The continuing growth of available online text documents makes research and applications of document summarization very important and consequently attracts many researchers. Vast number of available documents causes information overload and redundancy which makes it difficult to find and use information efficiently and effectively. Document summarization is one of the essential tools to overcome the part of this obstacle and therefore there is a raising need for new methods for tackling this urgently practical problem. Document summarization is the process of producing a generic or topic-focused compressed summary of a document or a set of documents sharing the same or similar topics by reducing document(s) in length while preserving the major semantic essence of the original document(s) (Mani I, 2001; Ricardo and Berthier, 1999). According to the number of documents to be summarized, the summary can be a single document or a multi-document. Single-document summarization can only distill one document into a shorter version, while on the contrary multi-document summarization can compress a set of documents. Multi-document summarization can be seen as an enhancement of single-document summarization and used for outlining the information contained in a cluster of documents. When used aside with document clustering, summarization techniques contribute to retrieving important and meaningful information from documents, and they have a wide range of applications in information management and retrieval (Wang at al., 2011). Since multi-document summarization combines and integrates the information across documents, it performs data synthesis and data mining. Based on the purpose, the summaries can be categorized into generic and query-based summaries. A generic summary condenses an overall gist of a document(s) content, whereas a query-based summary condenses only the content of a document(s) that are relevant to a provided query (Mani I, 2001). Another categorization of the text summarization systems is based on the methods used in the summarization algorithms. Roughly, there are two approaches for document summarization, supervised and unsupervised (Fattah and Ren, 2009). The supervised methods treat document summarization as a classification task of determining whether a sentence should be included in the summary or not. However, they depend upon the training samples, which are hard to obtain. The unsupervised methods generally employ clustering algorithms to score the sentences in the documents by combining a set of predefined features (Aliguliyev M, 2010; Cai and Li, 2011; Wang at al., 2008).

Recently, many generic document summarization methods using matrix factorization techniques have been proposed (Arora and Ravindran, 2008; Lee et al, 2003; Yeh et al, 2005; Lee et al, 2009; Wang at al., 2008; Mei et al., 2012; Ledeneva et al., 2011; Wang at al., 2008). These techniques can be jointly seen as a factor analysis description of input data exposed to different constraints. Even though they show significant similarities, due to different inner data handling and the type of the data analyses they offer, these methods can be practically categorized into low-rank factorization and clustering methods. An advantage of low rank approximations is that they have a great degree of flexibility but the features can be harder to interpret. While, clustering approaches extract features that are similar to actual data, making the results easier to interpret, on the other hand the binary assignments reduce flexibility.

Subsequently investigating pros and cons of a method able to directly combine the virtues of clustering and the flexibility of matrix factorization with application to the task of multi-document summarization is the main investigation objective of this work. In this paper, we propose a new unsupervised generic document summarization method based on Archetypal Analysis (AA).

The proposed method has the following advantages: (i) it is an unsupervised method; (ii) it is a language independent method; (iii) it is also a graph based method; (iv) in contrast to other factorization methods which extract prototypical, characteristic, even basic sentences, AA selects distinct (archetypal) sentences, thus induces variability and diversity in produced summaries; (v) the graph based methods require some kind of the sentence to sentence similarity matrix while the model-based methods use the term-sentence matrix document representation. Our approach can extract sentences by making the use of both types (graph-based and model-based) separably. It performs much better in term of effectiveness when the joint model of term-sentence and sentence-similarity matrix, namely the content-graph joint model is used; (vi) extracted sentence can be represented as convex combination of archetypal sentences, while the archetypes themselves are restricted to being very sparse mixtures of individual sentences and thus supposed to be more easily interpretable; and finally (vii) it readily offers soft clustering, i.e. simultaneous sentence clustering and ranking. To show the efficiency of the proposed approach, we compare it to other closely related summarization methods. We have used the DUC2004 and DUC2006 data sets to test our proposed method empirically. Experimental results show that our approach significantly outperforms the baseline summarization methods and the most of the state-of-the-art approaches.

The remainder of this paper is organized as follows: Section 2 describes related work regarding document summarization and the Archetypal Analysis. The details of the proposed summarization approach AASum are presented in Section 3, where we give an overview of the new approach, an illustrative example, discussions and relations to similar methods. Section 4 shows the evaluation and experimental results. Finally, we conclude in Section 5.

2 Related Work

2.1 Multi-Document Summarization

In recent years, algebraic methods, more precisely matrix decomposition approaches have become a key tool for document summarization. The typical approaches used in MDS spread from low rank approximations such as singular value decomposition (SVD) (Arora and Ravindran, 2008; Steinberger and Jezek, 2004), principal component analysis (PCA) (Lee et al, 2003), latent semantic indexing (LSI/LSA)(Yeh et al, 2005; Gong and Liu, 2001), non-negative matrix factorization (NMF) (Lee et al, 2009) and symmetric-NMF (Wang at al., 2008) to soft clustering approaches such as fuzzy K-medoids (Mei et al., 2012) and the EM-algorithm for clustering (Ledeneva et al., 2011) and hard assignment clustering methods such as K-means (Wang at al., 2008). Graph based methods can also be categorized as a decomposition methods as they are based on eigen decomposition which is closely related to the SVD.

Graph-based methods like LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004) model a document or a set of documents as a text similarity graph constructed by taking sentences as vertices and the similarity between sentences as edge weights. They take into account the global information and recursively calculate the sentence significance from the entire text graph rather than simply relying on unconnected individual sentences. These approaches were inspired by PageRank (Page et al., 1998) that has been successfully applied to rank Web pages in the Web graph. The recently proposed document-sensitive graph model (Wei et al, 2010) that emphasizes the influence of global document set information on local

sentence evaluation, is shown to perform better than other graph models for multi-document summarization task where MDS is modeled as single combined document summarization. Although those methods has shown as successful in covering relevance by calculating the principal or dominant eigenvector, they suffer from some fundamental limitations such as lack of diversity in produced summaries (Mei et al., 2010; Zhu et al, 2007), and topic drift handling (Richardson and Domingos, 2001). As these algorithms tend to ignore the influence of eigenvectors other than the largest one, the sentences related to topics other than a central one can be ignored, and thus creating the possibility for the inclusion of redundant sentences as well. This kind of summary cannot be considered as a generic one. A model presented in (Mei et al., 2010) automatically balances the relevance and the diversity of the top ranked vertices in a principled way. The most related model to DivRank is Grasshopper, which is a vertex selection algorithm based on the absorbing random walk (Zhu et al, 2007).

Latent Semantic Analysis (LSA) is an approach to overcome problems of multiple theme coverage in summaries by mapping documents to a latent semantic space, and has been shown to work well for text summarization. The document summarization method using LSA applies singular value decomposition (SVD) to summarize documents. This method decomposes term-document matrix into three matrices, U, D, and V. Starting from the first row of VT, the sentence corresponding to the column that has the largest index value with the right singular vector is extracted, to be included in the summary (Yeh et al, 2005; Gong and Liu, 2001). However, LSA has a number of drawbacks, namely its unsatisfactory statistical foundations. The EM-algorithm for clustering is utilized in work by (Bhandari et al, 2008) where document summarization is based on Probabilistic Latent Semantic Analysis (PLSA). The technique of PLSA assumes a latent lower dimensional topic model as the origin of observed term co-occurrence distributions, and can be seen as a probabilistic analogue to LSA. It has a solid statistical foundation, it is based on the likelihood principle, employs EM-algorithm for maximizing likelihood estimation and defines a proper generative model for data. PLSA allows classifying the sentences into several topics. The produced summary includes sentences from all topics, which made the generation of generic summary possible.

Automatic generic document summarization based on non-negative matrix factorization (Lee et al, 2009) is yet another successful algebraic method. This type of methods conduct NMF on the term-sentence matrix to extract sentences with the highest probability in each topic. NMF can also be viewed as a clustering method, which has many nice properties and advantages. Intuitively, this method clusters the sentences and chooses the most representative ones from each cluster to form the summary. NMF selects more meaningful sentences than the LSA-related methods, because it can use more intuitively interpretable semantic features and is better at grasping the innate structure of documents. As such, it provides superior representation of the subtopics of documents. The SNMF summarization framework, as an extension of (Lee et al, 2009), is based on sentence-level semantic analysis (SLSS) and symmetric non-negative matrix factorization SSNF. SLSS can better capture the relationships between sentences in a semantic manner and SSNF can factorize the similarity matrix to obtain meaningful groups of sentences. However SNMF is unable to define closeness to the cluster center and closeness to the sentences in the same cluster, therefore it is incapable of considered both in defining the subtopic-based feature. A fuzzy medoid-based clustering approach, as presented in (Mei et al., 2012) is an example of soft clustering methods for MDS. It is successfully employed to generate subsets of sentences where each of them corresponds to a subtopic of the related topic. This subtopic-based feature captures the relevance of each sentence within dif-

ferent subtopics and thus enhances the chance of producing a summary with a wider coverage and less redundancy.

In this work, we propose a new algebraic method based on archetypal analysis of the content-graph joint model. Archetypal analysis can be utilized to simultaneously cluster and rank sentences and content-graph joint model can better describe the relationships between sentences. Experimental results demonstrate the effectiveness of our proposed framework.

2.2 Archetypal Analysis

Archetypal Analysis (AA) as presented by Cutler and Breiman (1994) estimates each data point in a data set as a mixture of points of pure, not necessarily observed, types or archetypes. The archetypes themselves are restricted to being sparse mixtures of the data points in the data set, and lie on the data set boundary, i.e., the convex hull, see also Fig. 1. AA model can naturally be considered a model between low-rank factor type approximation and clustering approaches, and as such offers interesting possibilities for data mining. Since the coefficient vectors of archetypes locate in a simplex, AA readily offers soft clustering, probabilistic ranking, or classification using latent class models. So far, AA has found application in in different areas, e.g., in economics (Porzio et al., 2008), astrophysics (Chan, 2003), biology (Huggins et al., 2007) and recently in pattern recognition (Bauckhage and Thurau, 2009). The usefulness of AA model for feature extraction and dimensionality reduction for a large variety of machine learning problems taken from computer vision, neuro imaging, chemistry, text mining and collaborative filtering, is vastly presented in (Mørup and Hansen, 2012). For detailed explanation on numerical issues, stability, computational complexity and implementation of the AA we also refer to (Eugster and Leisch, 2009).

3 AASum - Archetypal Analysis based document Summarization

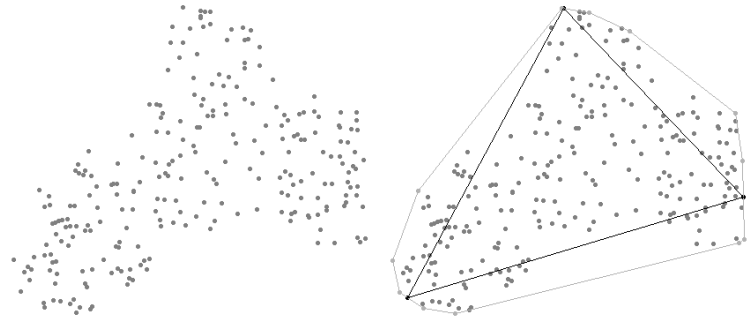
In this section we first present an overview of the archetypal analysis, following with detailed MDS problem statement and a new summarization method, called AASum. AASum employs the archetypal analysis for document summarization. An illustrative example, discussions and properties of the proposed method are also given.

3.1 Archetypal Analysis

Consider an n x m matrix X representing a multivariate data set with n observations and m variables. For given $k \ll n$ the archetypal problem is to decompose a given matrix X into stochastic matrices $H \in \mathbb{R}^{n \times k}$ and $W \in \mathbb{R}^{n \times k}$ as shown in Eq. 1

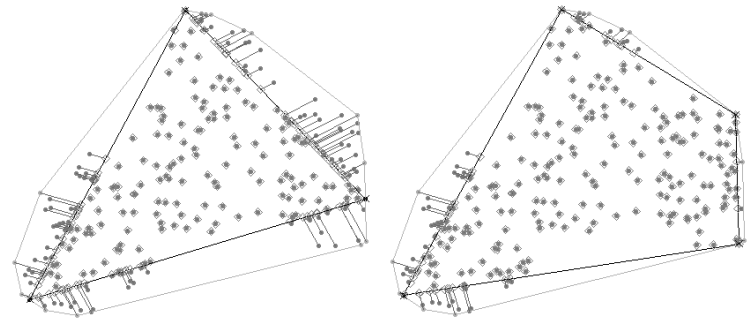
$$X \approx XWH \tag{1}$$

More exactly, the archetypal problem is to find two matrices W and H which minimize the residual sum of squares



(a) Data set in x,y plain

(b) convex hull and hull approximated with 3 archetypes



(c) point contributions to RSS

(d) convex hull and hull approximated with 4 archetypes and RSS visualization

Fig. 1 Archetypal analysis approximates the convex hull of a set of data. Increasing the number k of archetypes improves the approximation (b,d). While points inside an approximated convex hull can be represented exactly as a convex combination of archetypes, points on the outside are represented by their nearest point on the archetype hull (c,d). Suitable archetypes result from iteratively minimizing the residuals of the points outside of the hull (c,d). RSS stands for Residual Sum of Squares.

$$\begin{aligned} RSS(k) &= \|X - HY^T\|^2 \text{ with } Y = X^T W \\ s.t. \sum_{j=1}^{k} H_{ij} &= 1, H_{ij} \geq 0; \sum_{i=1}^{n} W_{ij} = 1, W_{ij} \geq 0 \end{aligned} \tag{2}$$

The constraint $\sum_{i=1}^{k} H_{ij} = 1$ together with $H_{ij} \geq 0$ enforces the feature matrix Y to be a convex combination (i.e., weighted average) of the archetypes while the constraints $\sum_{i=1}^{m} W_{ij} =$

1 and $W_{ij} \geq 0$, require that each archetype is a meaningful combination of data points. $\|\cdot\|^2$ denotes the Euclidean matrix norm.

The description of archetypal analysis given in Eq. 2 defines the foundation of the estimation algorithm first presented in (Cutler and Breiman, 1994). It alternates between finding the best H for given archetypes Y and finding the best archetypes Y for given H; where at each step many convex least squares problems are solved until the overall RSS is reduced successively.

The inclusive framework for archetypal analysis in step by step description is the following:

Given the number of archetypes k:

1. Pre-processing: scale data.
2. Initialization: initialize W in a way the constrains are satisfied to calculate the starting archetypes Y
3. Repeat while a stopping criterion is not met, i.e., stop when RSS is small enough or the maximum number of iteration is reached:
   3.1 Find best H for the given set of archetypes Y, i.e, solve n convex least squares problems, where $i = 1, ..., n$

$$\min_{H_i} = \frac{1}{2}\|X_i - YH_i\|^2 \text{ s.t. } \sum_{j=1}^{k} H_{ij} = 1, H_{ij} \geq 0.$$

   3.2 Recalculate archetypes $\hat{Y}$ by solving system of linear equations $X = H\hat{Y}^T$.
   3.3 Find best W for the given set of archetypes $\hat{Y}$, i.e, solve k convex least squares problems where $j = 1, ..., k$

$$\min_{W_j} = \frac{1}{2}\|\hat{Y}_j - XW_j\|^2 \text{ s.t. } \sum_{i=1}^{n} W_{ij} = 1, W_{ij} \geq 0.$$

   3.4 Recalculate archetypes $Y = XW$
   3.4 Recalculate RSS.
4. Post-processing: rescale archetypes

Initialization: Cutler and Breiman point out that some attention should be given in choosing initial mixtures that are not too close together because this can cause slow convergence or convergence to a local optimum. To ensure Breiman's point on choosing initial mixtures (archetypes) we use the following method. The method proceeds by randomly selecting a data point as archetype and selecting subsequent data points (archetypes) the furthest away from already selected ones noted as x_i. As such a new data point is selected according to

$$a^{new} = \arg\max_i \left\{ \sum_j \|x_i - x_j\|, j \in C \right\} \tag{3}$$

where $\|\cdot\|$ is a given norm and C is a set of indeces of current selected points.

Convergence: Cutler and Breiman (1994) show that the algorithm converges in all cases, but not necessarily to a global minimum. They also note that, alike many alternating optimization algorithms, their algorithm results in a fixed point of an appropriate transformation, but there is no guarantee that this will be a global minimizer of RSS. For further details on convergence see (Cutler and Breiman, 1994).

$$\overset{X_{nmf}}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}} \approx \overset{N}{\begin{bmatrix} 0 & 3.7423 \\ 5.9965 & 2.8865 \\ 11.9938 & 2.0307 \\ 17.9910 & 1.1749 \end{bmatrix}} \times \overset{M}{\begin{bmatrix} 0.5384 & 0.5765 & 0.6146 \\ 0.2673 & 0.5345 & 0.8018 \end{bmatrix}} = \overset{\tilde{X}_{nmf}}{\begin{bmatrix} 1.0004 & 2.0004 & 3.0004 \\ 4.0000 & 5.0000 & 6.0000 \\ 7.0000 & 8.0000 & 9.0000 \\ 10.0000 & 11.0000 & 12.0000 \end{bmatrix}}$$

(a) NMF decomposition results

$$\overset{X_{aa}}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}} \approx \overset{X_{aa}}{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}} \times \overset{W}{\begin{bmatrix} 0 & 0.9545 \\ 0.1116 & 0.0455 \\ 0.8884 & 0 \end{bmatrix}} \times \overset{H}{\begin{bmatrix} 0 & 0.5101 & 1 \\ 1 & 0.4899 & 0 \end{bmatrix}} = \overset{\tilde{X}_{aa}}{\begin{bmatrix} 1.0455 & 1.9856 & 2.8884 \\ 4.0455 & 4.9856 & 5.8884 \\ 7.0455 & 7.9856 & 8.8884 \\ 10.0455 & 10.9856 & 11.8884 \end{bmatrix}}$$

(b) AA decomposition results

Fig. 2 Decomposition examples from NMF and AA .

Example 1. In order to show AA and another well know matrix factorization method in use we illustrate the following example. In NMF the non-negative matrix X is decomposed into two nonnegative matrices, N and M, as shown in Fig. 2a. In AA the matrix X is decomposed into two stochastic matrices, W and H as shown in Fig. 2b. In contrast to NMF, AA decomposes an input sparse matrix into two very sparse stochastic matrices. Fig. 3 shows this property of the AA. Here, the sparseness of a matrix is the number of zero elements divided by the total number of elements of the matrix. The non-negative matrices X in Fig. 3 were a randomly generated n-by-n matrices, and the values of n were set to 20, 50, 80, 100 and 200.

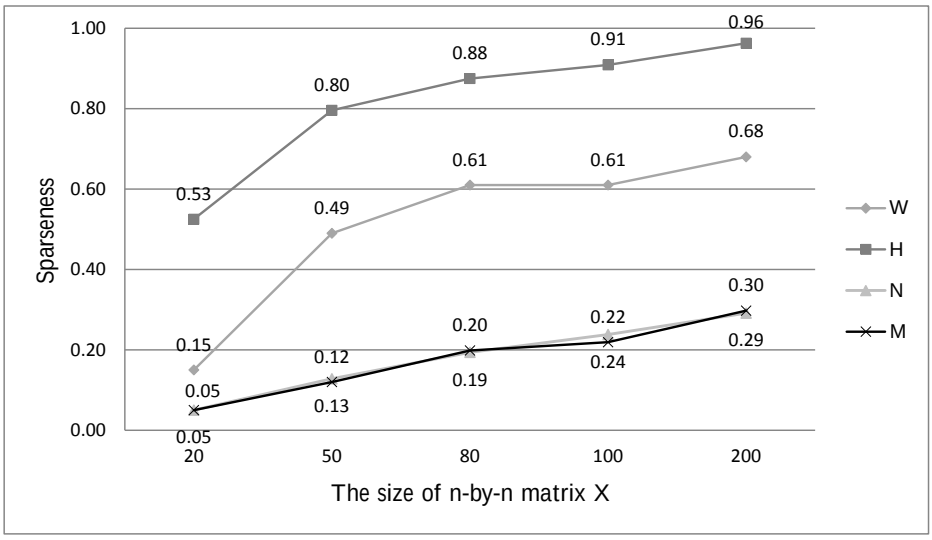


Fig. 3 Comparison of sparseness between AA and NMF decomposition.

Text summarization has four important aspects. The first aspect is relevancy which assures that a summary contains the most important information. The selected sentences have to be closely relevant to the main content of the corpus. The second one is content coverage. A summary should cover as many as possible of the important aspects of the documents and in this way should minimize the information loss in summarization process. Another important aspect is diversity which promotes the idea that a good summary should be brief and should contain as few redundant sentences as possible, that is, two sentences with similar meaning should not be both selected to form the summary. Practically, the diversity requirement in summarization can productively minimize redundancy in produced summaries. The last aspect is the length of a summary which is usually user defined. Optimizing all these properties is a severe task and is an example of a general summarization problem. Our objective is to extract a small subset of sentences from a collection of documents such that the created summary fulfill the above requirements. In our study, this goal has been reached with using the archetype analysis. To apply the AA to the sentence-extraction-based document summarization we use the joint model of term-sentence and sentence-similarity matrix, namely the *content-graph joint model*.

Let a document corpus be separated into a set of sentences $D = \{s_1, s_2, ..., s_n\}$, where n denotes the number of sentences, s_i denotes ith sentence in D. In the interest of forming the term-sentence and sentence-similarity matrices each of the sentences should be presented as a vector. The vector space model is the most known representation scheme for textual units. It represents textual units by counting terms or sequence of terms. Let $T = \{t_1, t_2, ..., t_m\}$ represent all the distinct terms occurring in the collection, where m is the number of different terms. The standard vector space model (VSM) using the bag of the words approach represents the text units of a corpus as vectors in a vector space. Traditionally, a whole document is used as a text unit, but in this work we use only sentences. Each dimension of a vector corresponds to a term that is present in the corpus. A term might be, for example, a single word, N-gram, or a phrase. If a term occurs in a sentence, the value of that dimension is nonzero. Values can be binary, frequencies of terms in the sentence, or term weights. Term weighting is used to weight a term based on some kind of importance. The most often used measure is the raw frequency of a term, which only states how often the term occurs in a document without measuring the importance of that term within the sentence or within the whole collection. Different weighting schemes are available. The most common and popular one is the term frequency inverse sentence frequency (tf-isf) weighting scheme. It combines local and global weighting of a term. The local term weighting measures the significance of a term within a sentence:

$$tf_{ik} = freq_{ik} \tag{4}$$

where $freq_{ik}$ is the frequency of term t_k in sentence s_i. With this formula, terms that occur often in a sentence are assessed with a higher weight. The global term weighting or the inverse sentence frequency isf measures the importance of a term within the sentence collection:

$$isf_{ik} = log(\frac{n}{n_k}) \tag{5}$$

where n denotes the number of all sentences in the corpus, and n_k denotes the number of sentences that term t_k occurs in. This formula gives a lower isf value to a term that occurs in

many sentences, and in this way it favors only the rare terms since they are significant for the distinction between sentences. As a result the $tf\text{-}isf$ weighting scheme can be formulated as:

$$w_{ik} = tf_{ik} \times isf_{ik} = freq_{ik} \times log(\frac{n}{n_k}) \tag{6}$$

here the weight w_{ik} of a term t_k in a sentence s_i is defined by the product of the local weight of term t_k in sentence s_i and the global weight of term t_k. A very popular similarity measure is the cosine similarity which uses the weighting terms representation of the sentences. According to the VSM the sentence s_i is represented as a weighting vector of the terms, $s_i = [w_{i1}, w_{i2}, ..., w_{im}]$, where w_{ik} is the weight of the term t_k in the sentence s_i. This measure is based on the angle α between two vectors in the VSM. The closer the vectors are to each other the more similar are the sentences. The calculation of an angle between two vector $s_i = [w_{i1}, w_{i2}, ..., w_{im}]$ and $s_j = [w_{j1}, w_{j2}, ..., w_{jm}]$ can be derived from the Euclidean dot product:

$$(s_i, s_j) = |s_i| \cdot |s_j| \cdot \cos\alpha \tag{7}$$

This states that the product of two vectors is given by the product of their norms (in spatial terms, the length of the vector) multiplied by the cosine of the angle α between them. Given Eq.7 the cosine similarity is therefore:

$$sim\,(s_i, s_j) = \cos\alpha = \frac{(s_i, s_j)}{|s_i| \cdot |s_j|} = \frac{\sum_{l=1}^{m} w_{il} w_{jl}}{\sqrt{\sum_{l=1}^{m} w_{il}^2 \cdot \sum_{l=1}^{m} w_{jl}^2}}, i, j = 1, 2, \cdots n. \tag{8}$$

The sentence similarity matrix describes a similarity between sentences presented as a points in Euclidean space. Columns and rows are sentences while their intersection gives the similarity values of corresponding sentences calculated with Eq.8.

The term-sentence matrix is a mathematical matrix that describes the frequency of terms that occur in sentences from a collection of documents. In this matrix, rows correspond to terms and columns to sentences from the collection of documents. A term-frequency vector for each sentence in the document is then constructed using Eq.6.

The content-graph joint model is constructed from the sentence similarity matrix and the term-sentence matrix. Cohn and Hofmann (2000) have demonstrated that building a joint model of document contents and connections produces a better model than that built from contents or connections alone. Let the number of sentences in the documents be m and the number of terms n. Then T denotes the $n \times m$ term-sentence matrix and a sentence to sentence similarity matrix may also be represented as a vector space, defining an $n \times n$ matrix A. The straightforward way to produce such a joint model is to calculate the matrix product TA, and then to factor the product via AA. In matrix notation,

$$[TA]_{n \times m} = [T]_{n \times m} \times [A]_{m \times m} \tag{9}$$

The content-graph joint model provide a methodical way of combining information from both the terms and sentence similarity connection structure present in the corpus.

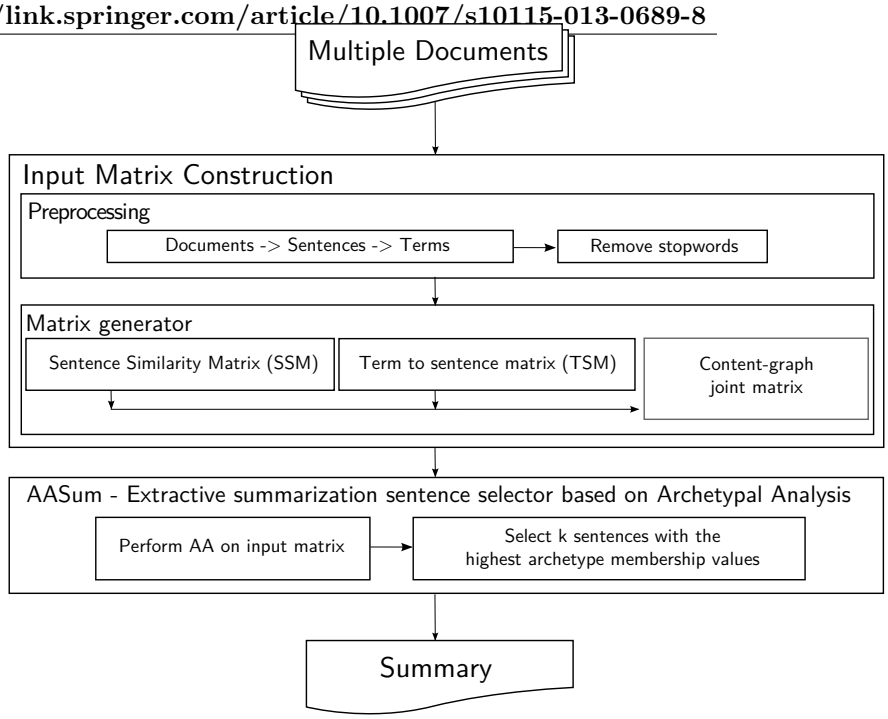


Fig. 4 MDS method using AA

3.3 Generic document summarization by AASum

In this subsection, a method for generating multi-document summary by selecting sentences using AA is presented. Modeling texts as graphs implies having as their vertices text segments and as their links information on how these nodes relate to each other. For summarization purposes, graph metrics signal the importance of a text segment. In this sense AASum is an enhanced version of a typical graph based model, since it makes use of the content-graph joint matrix. Informally, we can look at the content-graph representation as a graph where a sentence is connected to sentences that have $[T \times A]$ term distribution.

We give a full explanation of the method in Fig 4. Here k denotes the number of sentences to be extracted. The main idea of the method is simple: sentences are soft-clustered into archetypes in order to produce the sentence ranking where the top ranked ones are then sequentially extracted, until the length constraint (l sentences) is reached.

The framework of the proposed unsupervised multi-document summarization method AASum consists of the following steps:

1. Decompose the document D into n individual sentences
2. Perform the preprocessing.
   i Tokenize the sentences into words.
   ii Remove the stopwords
3. Construct the input matrix X.

i Produce the sentence similarity matrix A using Eq.8.
ii Generate the term to sentence occurrence matrix T by using Eq.6.
iii Return the matrix product of A and T using the Eq.9.

4. Perform AA on matrix X.
   i Estimate the decomposition matrices H,W and XW using the AA algorithm given in Sec. 3.1.
   ii For each archetype i calculate its significance i.e. the sum of values in corresponding column of the matrix XW, $Sa_i = \sum_{j=1}^{m} XW_{j,i}$.
   iii Sort the archetypes in decreasing order of significance, i.e. order the rows of matrix W based on values of Sa_i.
   iv Eliminate ϵ archetypes with lowest significance and return the result.
5. Select l sentences with the highest archetype membership value from the most significant archetypes.
   i Start with the most significant archetype (the first row of the row-sorted matrix W) and extract the sentence with highest value in this row. Then continue with second most significant archetype (the second row of W) and so on. That is, sentences with highest archetype membership values in each archetype are selected one by one and if the summary length is not met then the extraction step continues with the second highest values in each archetype, and so forth.
   ii Each selected sentence is compared to previously selected ones and if there is a significant similarity between them, the newly selected sentence is not included in the summary.

Here, ϵ denotes the number of the least significant archetypes. In the above algorithm, the fourth and fifth steps are the key steps. Our purpose is to cluster sentences into archetypes and afterward extract the sentences with the highest archetype membership weights. Since each sentence contributes to the identification of every single archetype then each sentence might have different values in the rows of the matrix W. Hence, the same sentence s can have higher membership value in one and lower membership value in the other archetype. But considering that our goal is to identify the "best summary" sentences our method will select the sentence s as a summary only if it has a high archetype membership value in one of the significant archetypes. By the fourth step, the salient sentences are more likely to be clustered into archetypes with high significance. Because the sentences with the higher membership values are ranked higher, the sentences extracted by the fifth step are the most representative ones. Another point to mention is that the facts with higher weights appear in a greater number of sentences, therefore archetypal analysis clusters such fact-sharing sentences in the archetype with higher weight. Thus, the fifth step in the above algorithm starts the sentence extraction with the largest archetype to ensure that the system-generated summary first covers the facts that have higher weights. In this way our method optimizes the two important aspect of the summarization, namely the relevance and the content coverage. The last important function of these two steps is diversity optimization. This is to some extent provided by the definition of archetypal analysis which clusters sentences into distinct archetypes. Nevertheless, in order to more effectively remove redundancy and increase the information diversity in the summary, we use a greedy algorithm presented in the last step (5.ii) of above algorithm. In the following subsection we present the usage of AASum on an illustrative example.

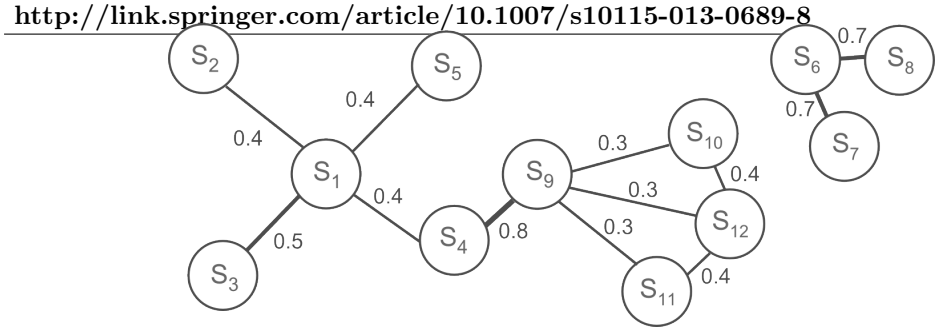


Fig. 5 Sentence similarity graph

Table 1 Results of Archetype Analysis on the illustrative example

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	$\sum R_i$
H_1	0	0.7944	0.7944	1	0.7944	0	0	0	0	0.8785	0.8785	0.6834	
H_2	0.0711	0.1798	0.1798	0	0.1798	1	1	1	0	0.0235	0.0235	0.0240	
H_3	0.9288	0.0257	0.0257	0	0.0257	0	0	0	1	0.0979	0.0979	0.2925	
W_1^T	0	0.1239	0.1239	0.2493*	0.1239	0	0	0	0	0.1460	0.1460	0.0877	
W_2^T	0	0.0155	0.0155	0	0.0155	0.3240*	0.3100	0.3100	0	0	0	0	
W_3^T	0.4187*	0	0	0	0	0	0	0	0.5313*	0	0	0.0509	
XW_1^T	0.2734	0	0	0	0	0	0	0	0.3134	0.0438	0.0438	0.1460	0.8206
XW_2^T	0.0187	0	0	0	0	0.4453	0.2226	0.2226	0	0	0	0	0.9093
XW_3^T	0	0.1674	0.1674	0.6344	0.1674	0	0	0	0.0152	0.1848	0.1848	0.1593	1.6813

3.4 An illustrative example

In order to demonstrate the advantages of AA as the method of simultaneous sentence clustering and ranking, a simple example is given in Fig. 5. We present the synthetic data set as an undirected sentence similarity graph, where nodes denote sentences and edges represent similarity between connected nodes. Looking at the data directly, one can observe two clusters of sentences, where $\{S_1, S_9\}$ are the central sentences of the first and the S_6 of the second cluster. One can also argue that there is a topic drift in the first cluster occurring in the neighborhood of S_4.

By setting Z=3 we obtain matrices H, W and XW estimated by AA as shown in Table 1. Decomposed matrices H and W can be interpreted as clustering and ranking outputs, respectively. Extracted archetypes, three of them, are in fact data-driven extreme values. In summarization, these extreme values are the archetypal sentences which are outstanding, positively and/or negatively. For interpretation, we identify the archetypal sentences as different types with different degree of potentially "good" and "bad" summary sentences, and set the observations in relation to them. In order to sort archetypes according to their significance we first calculate the sum of the each row of the matrix XW^T, and then we order the archetypes based on the row sums. From the last column of Table 1, it can be seen that $Archetype_3$ is the most significant sentence archetype and it can be seen as "very-good" archetype while $Archetype_1$ is the least significant and it can be considered as "bad" from which no sentence should be extracted

into the summary. From H_2 in Table 1 it can be seen that $\{S_6, S_7, S_8\}$ belong to the second cluster, the good archetype, while the rest of the sentences belong to other two archetypes with various values of membership. W_3^T, the "very-good" archetype, shows that $\{S_9, S_1\}$ have the highest ranking values, therefore they should be extracted into the resulting summary. From W_2^T, the "good" archetype, it is obvious that S_6 has the highest ranking value in this cluster and it should also be extracted to the resulting summary. From W_1^T it can be seen that S_4 is the most salient sentence in "bad" archetype, nevertheless this can be also interpreted as the point of the topic drift in the first cluster of the original data set. This example shows that output matrices produced by AA describe the data structure well and in various ways, i.e., H reflects the clustering into archetypes and W the rank within each cluster. It is a non-trivial problem to choose the best number of archetypes to be estimated by AA. But since we only select one representative sentence from each archetype starting from the most significant and not including ϵ the least significant ones, then the number of archetypes k may be set to be close to the number of sentences to be extracted plus the number ϵ.

3.5 Discussions and Relations

Since various matrix decomposition methods such as PCA/SVD, k-means and NMF have been successfully employed in MDS task, it is reasonable and in interest of the reader to investigate the connection of those factorization methods. Expressed in terms of optimization problems, one can state that PCA/SVD, NMF, k-means and AA are special cases of a more general problem P_G. Given any matrix $X \in \mathbb{R}_+^{m \times n}$ and any positive integer p, the problem P_G can be stated as follows. Find the best nonnegative factorization $P \approx L_1 L_2$ (with $L_1 \in \mathbb{R}_+^{m \times p}, L_2 \in \mathbb{R}_+^{p \times n}$) i.e.

$$(L_1 L_2) = \arg\min_{L_1 L_2} \|P - L_1 L_2\|^2 \tag{10}$$

Thus, the presented decomposition methods can be ordered according to the specificity of constants involved in the problem. Here we summarize the relations of methods and sort them in the decreasing order:

1. AA:

$$(WH) = \arg\min_{W,H} \|X - XWH\|^2$$
s.t. W is stochastic
H is stochastic

2. k-means

$$(WH) = \arg\min_{W,H} \|X - WH\|^2$$
s.t. W is stochastic
H is binary

3. NMF:

$$(WH) = \arg\min_{W,H} \|X - WH\|^2$$
s.t. W is nonnegative
H is nonnegative

4. PCA/SVD :

$$(WH) = \arg\min_{W,H} \|X - WH\|^2$$
s.t. $W^T W = I$

NMF, k-mean and SVD have been shown as successful decomposition method for MDS, ergo one can expect similar or even better results from AA. This claim is based on the presented formulation where AA is ordered as the most special instance of the given optimization

Table 2 Description of data sets

	DUC 2004	DUC 2006
Number of clusters	50	50
Number of documents per cluster	10	25
Average number of sentences per cluster	257.48	690.78
Total number of documents in the corpus	500	1250
Total number of sentences in the corpus	12,874	34,539
Summary length	665 bytes	250 words

problems. Supporting evidences can be found in next section where we compare AASum with other decomposition based summarization methods.

4 Experiments

In this section, we conduct experiments on two DUC data sets to evaluate the effectiveness and possible positive contributions of the proposed method compared with other existing summarization systems.

4.1 Experimental data and evaluation metric

We use the DUC2004 and DUC2006 data sets to evaluate our proposed method empirically, where benchmark data sets are from DUC (http://duc.nist.gov) for automatic summarization evaluation. DUC2004 and DUC2006 data sets consist of 50 topics. Each topic of DUC2004 and DUC2006 includes 10 and 25 documents, respectively. Table 2 gives a brief description of the data sets. The task is to create a summary of no more than 650 bytes and 250 words, respectively. In those document data sets, stop words were removed using the stop list provided in (ftp://ftp.cs.cornell.edu/pub/smart/english.stop) and the terms were stemmed using the Porter's scheme (http://www.tartarus.org/martin/PorterStemmer/), which is commonly used algorithm for word stemming in English.

The summarization evaluation methods can be divided into two categories: intrinsic and extrinsic (Mani I, 2001; Fattah and Ren, 2009). The intrinsic evaluation measure the quality of summaries directly (e.g., by comparing them to ideal summaries). The extrinsic methods measure how well the summaries help in performing a particular task (e.g., classification). The commonly used technique to measure the interjudge agreement and to evaluate extracts is ROUGE metric. In our experiments, we used for evaluation the Recall Oriented Understudy for Gisting Evaluation (ROUGE) evaluation package (Lin C-Y, 2004), which compares various summary results from several summarization methods with summaries generated by humans. ROUGE is adopted by DUC as the official evaluation metric for text summarization. It has been shown that ROUGE is very effective for measuring document summarization. It measures how well a machine summary overlaps with human summaries using N-gram co-occurrence statistics, where an N-gram is a contiguous sequence of N words. Multiple ROUGE metrics are defined according to different N and different strategies, such as ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. The ROUGE-N measure compares N-grams of two summaries, and counts the number of matches. This measure is computed by formula (Lin

Table 3 AASum methods comparison in input matrix construction phase on DUC2004 and DUC2006. Remark: * indicates the best results in this set of experiments

	DUC2004		DUC2006	
Summarizers	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
AASum-W1	0.3605	0.0683	0.3334	0.0441
AASum-W2	0.3706	0.0871	0.4239	0.0908
AASum-W3	0.4115*	0.0934*	0.4291*	0.0944*

C-Y, 2004)

$$\text{ROUGE-N} = \frac{\sum_{S \in Summ_{ref}} \sum_{\text{N-gram} \in S} Count_{match}(\text{N-gram})}{\sum_{S \in Summ_{ref}} \sum_{\text{N-gram} \in S} Count(\text{N-gram})} \tag{11}$$

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. Here, we report the mean value over all topics of the recall scores of ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-SU4 (skip-bigram plus unigram) (Lin and Hovey, 2004).

4.2 Input matrix selection and it's impact on summarization

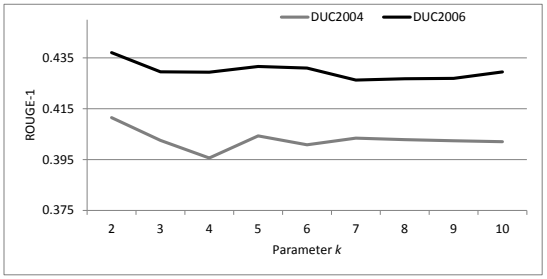
Actually, many summarization methods either directly perform on the terms by sentences matrix, such as the LSA and NMF, or they perform on sentence similarity matrix, which are also known as graph based methods such as LexRank and DSQ. Note that the two types are implemented as baseline systems in our experiments.
In the experiments, we compare the AASum method's summarization results with respect to the input matrix type. Depending on a type of the input matrix we apply AA in 3 different ways during the summarization process. Each of the way, denoted as AASum-W1, AASum-W2, AASum-W3 is discussed below. The AASum-W1, the content based method, performs on the term by sentence matrix formed by Eq.6. The AASum-W2, the graph based method, performs on the sentence by sentence similarity matrix constructed by Eq.8. The AASum-W3, the content-graph method, performs on the joint matrix of later two ones and it is obtained by using Eq.9. Informally, we can look at the graph-content representation as saying that a sentence is connected to sentences that have $[T * A]$ term distribution. In order to better understand the results, we use Table 3 to illustrate the comparison. The results clearly show that our method performs best on the content-graph input matrix. This is due to fact that the graph-content representation better describes the sentence relations.

4.3 Impact of the archetype algorithm's initialization on summarization

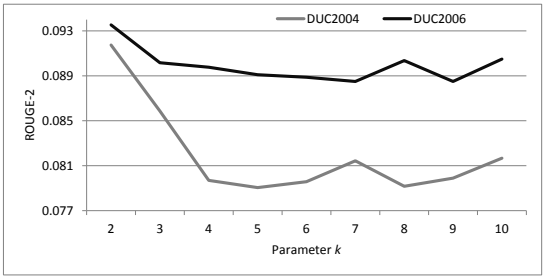
This section investigates whether the summarization outcome depends on the initialization of matrices W and H. As noted in section 3.1, one can simply initialize the matrix W to data points selected at random without replacement from the input data. Initialization process then

Table 4 Impact of the initialization of the archetype algorithm on summarization

	DUC2004			DUC2006		
Initialization	ROUGE-1	ROUGE-2	ROUGE-SU	ROUGE-1	ROUGE-2	ROUGE-SU
random	0.3990	0.0919	0.1488	0.4358	0.0873	0.1824
f-away	0.4057	0.0908	0.1510	0.4361	0.0881	0.1855
	0.0067	0.0011	0.0022	0.0003	0.0008	0.0031



(a) ROUGE-1 vs k on DUC2004 and DUC2006



(b) ROUGE-2 vs k on DUC2004 and DUC2006

Fig. 6 Impact of archetype number on summarization

continues with computing H and XW given the W. Let us name this initialization method as *random*. Another initialization method presented in section 3.1 is based on the idea of sequential archetype selection which are furthest away from each other. Let us name this second method as the *f-away*. In order to experimentally demonstrate the impact of initialization we designed the following experiment. We run AA algorithm sequentially 100 times with each of initialization methods. Then, in a very straightforward way, the average ROUGE scores over all runs are computed and compared. Results, presented in Table 4, suggest that the summarization outcome is not sensitive to the initialization method.

4.4 Impact of the number of archetypes

This problem is the same as the problem of choosing the number of components in other matrix decomposition approaches and there is no rule for defining the correct number of archetypes k. A simple approach for choosing the value of k is to run the algorithm for different numbers of k where the selection criteria should be the maximization of the summary evaluation outcomes. In previous experiments, the archetype number k is set to be close to the number of sentences to be extracted plus the number ϵ. The ϵ is the number of the least significant archetypes which are not used in the final sentence selection. To further examine how the number of archetypes influences the summarization performance, we conduct the following additional experiments by varying k. We gradually increase the value of k, in the range from 2 to 10 and the results show that increasing the number of extracted archetypes do not necessary increases the summarization

Table 5 Summarization systems

System ID	Description
AASUM	Our method
LSA	Matrix decomposition
NNF	Matrix decomposition
SNMF	Clustering, Matrix decomposition
SumCR-G	Clustering, Sub-topic
LexRank	Graph-based
DrS-G, DrS-Q	Document sensitive graph-based
DivRank	Graph-based, relevance and diversity balanced method
Human	Best human performance provided by DUC
System	Top few systems from DUC
Baseline	The baseline system used in DUC

Table 6 General evaluation of the AASum on the DUC2004 and DUC2006 datasets

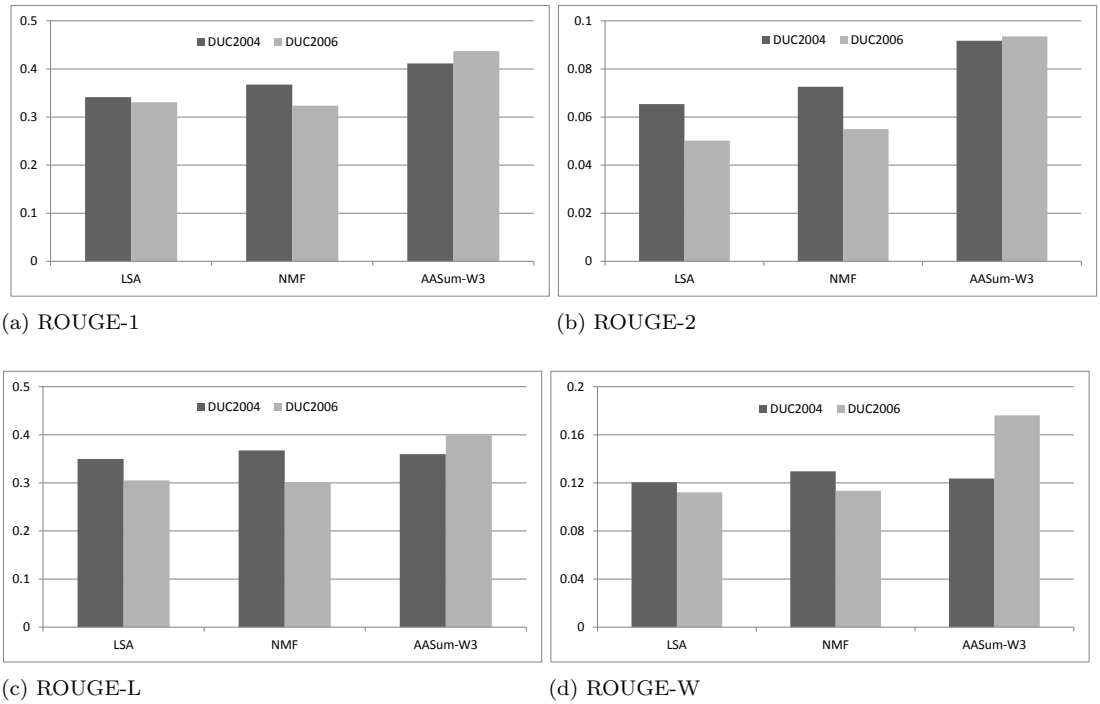
	ROUGE	DUC 2004	DUC 2006
Recall	1	0.4115 [0.3976-0.4255]	0.4291 [0.4220-0.4358]
	2	0.0934 [0.0838-0.1030]	0.0944 [0.0904-0.0985]
	L	0.3434 [0.3305-0.3562]	0.3908 [0.3860-0.3959]
	W	0.1180 [0.1134-0.1225]	0.3830 [0.3781-0.3883]
	SU	0.1376 [0.1280-0.1469]	0.1680 [0.1638-0.1723]
Precision	1	0.3989 [0.3837-0.4147]	0.4002 [0.3933-0.4066]
	2	0.0908 [0.0809-0.1007]	0.0904 [0.0865-0.0945]
	L	0.3330 [0.3199-0.3476]	0.3728 [0.3673-0.3782]
	W	0.2049 [0.1964-0.2139]	0.3653 [0.3600-0.3708]
	SU	0.1298 [0.1198-0.1400]	0.1532 [0.1488-0.1577]
F-measure	1	0.4049 [0.3907-0.4195]	0.4138 [0.4069-0.4202]
	2	0.0921 [0.0824-0.1018]	0.0923 [0.0884-0.0964]
	L	0.3379 [0.3249-0.3517]	0.3812 [0.3763-0.3864]
	W	0.1497 [0.1437-0.1554]	0.3736 [0.3686-0.3789]
	SU	0.1333 [0.1234-0.1428]	0.1597 [0.1556-0.1640]

performance. Fig. 6 plots the ROUGE-1 and ROUGE-2 curves of our AA based approach on the DUC2004 and DUC2006 datasets.

4.5 Comparison with related methods

We first report the standalone performance results of the proposed method in Table 6 where the mean value as well as 95% confidence interval over all topics of the recall, precision and f-measure scores of ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-W are reported.

Then we compare the proposed AASum with two most relevant methods, LSA and NMF. As indicated in (Lee et al, 2009), LSA and NMF are two competing matrix decomposition techniques for the task of MDS. From Fig. 7 we can see that NMF shows better perfomance than LSA. This is in consistency with results reported in (Lee et al, 2009) and it can be mainly contributed to the property of NMF to select more meaningful sentences by using the



(a) ROUGE-1 (b) ROUGE-2

(c) ROUGE-L (d) ROUGE-W

Fig. 7 Overall summarization performance on DUC2004 and DUC2006 data

more intuitively interpretable semantic features and by better grasping the innate structure of documents. Our proposed approach shows even better performance, see Fig. 7. This is because it uses the archetypal analysis to detect the archetypal structure which can cluster and rank sentences more effectively than above-mentioned approaches. Fig. 7 gives the improvements of AASum with respect to LSA and NMF, where it can be seen that AASum performs consistently much better than the other two approaches. Since both LSA and NMF are matrix factorization methods the improvement of AASum compared with them can be also attributed to AA's ability to combine the clustering and the matrix factorization.

In addition to these two methods, we compare AASum with some other approaches, see Table 5. Although there are, for each year, more than 30 systems that have participated in DUC competition, here we only compare with the top few systems. The advantages of our approach are clearly demonstrated in Table 7 and Table 8. It produces very competitive results, which apparently outperforms many of the methods in both years. More important, it is ahead of the best system in DUC2006 on ROUGE-1, and ranks among the bests in DUC2004. Note that in our present research the position of a sentence in the document is not studied yet. However, the position feature has been used in all the participating systems as one of the most significant features (Mei et al., 2012). Notice also that all the results of AASum are produced based on a simple similarity measure, and the query information is only incorporated in a straightforward way.

Table 7 Evaluation of the methods on the DUC2004 dataset. Remark: "-" indicates that the method does not officially report the results

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3242	0.0641	-
Best-Human	0.4182*	0.1050*	-
System-65	0.3822	0.0921	0.1332
System-35	0.3708	0.0834	0.1273
SNMF	-	0.0840	0.1266
SumCR-G	-	0.0965	0.1364
LexRank	0.3784	0.0857	0.1312
DrS-G	0.3752	0.0872	0.1290
AASum-W3	0.4115	0.0934	0.1376*

Table 8 Evaluation of the methods on the DUC2006 dataset.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3208	0.0527	0.1041
Best-Human	-	0.1036*	0.1683*
System-24	0.4102	0.0951	0.1546
System-12	0.4049	0.0899	0.1476
SNMF	0.3955	0.0855	0.1398
SumCR-G	-	0.0906	0.1437
LexRank	0.3899	0.0856	0.1394
DsR-Q	0.3955	0.0899	0.1427
AASum-W3	0.4291*	0.0944	0.1680

We believe that AASum has the potential to achieve further improvements in its performance on general and task-focused summarization by making use of the position feature and by incorporating sophisticated methods to make use of the query information in a more effective way.

5 Conclusion and future work

The main contributions of the paper are the following:

(i) The paper presents a document summarization method which extracts significant sentences from the given document set while reducing redundant information in the summaries with the coverage of topics of document collection.
(ii) Document summarization is formalized as the Archetypal Analysis problem that takes into account relevance, information coverage, diversity and the length limit.
(iii) The paper also shows how AA can be used for simultaneously sentence clustering and ranking.
(iv) This paper has showed that AASum performs much better in terms of effectiveness when the joint model of term-sentence and sentence-similarity matrix, namely the content-graph joint model is used.

(v) This paper has found that AASum is an effective summarization method. Experimental results on the DUC2004 and DUC2006 datasets demonstrate the effectiveness of the proposed approach, which compares well to most of the existing matrix decomposition methods in the literature.

We believe that in the future the performance of AASum would possibly be further improved. There are many potential directions for improvements of AASum such as: (1) in the general summatization task AASum has not made use of the sentence position feature; (2) in the query-based summarization, it has not employed any kind of the query processing techniques; (3) instead of using a semantic similarity, AASum currently only uses a simple similarity measure; (4) in the presented work AASum rather than truly summarizing multiple documents it treats the problem of MDS as a summarization of a single combined document; (5) another possible enhancement can be reached by introducing the multi-layered graph model that emphasizes not only the sentence to sentence and sentence to terms relations but also the influence of the under sentence and above term level relations, such as n-grams, phrases and semantic role arguments levels.

References

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

Arora R, Ravindran B (2008) Latent Dirichlet Allocation and Singular Value Decomposition Based Multi-document Summarization. In: *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM, IEEE Computer Society, pp 713-718.

Bauckhage C, Thurau C (2009)Making Archetypal Analysis Practical In: *Proceedings of Pattern Recognition 31st DAGM Symposium,*, LNCS,Springer pp 272-281.

Bhandari H, Shimbo M, Ito T, Matsumoto Y (2008) Generic text summarization using probabilistic latent semantic indexing In: *Proceedings of the 3rd International Joint Conference on Natural Langugage Proceeding 2008*, pp 133-140.

Cai X, LI W(2011) A spectral analysis approach to document summarization: Clustering and ranking sentences simultaneously *Inf. Sci.*, 181(18):3816–3827

Chan B-H-P. (2003) Archetypal Analysis of Galaxy Spectra. *Monthly Notices of the Royal Astronomical Society*, 338(3):790-795

Cohn A-D, Hofmann T. (2000) The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity. In: *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000*, pp 430-436.

Cutler A, Breiman L, (1994) Archetypal analysis, *Technometrics* , 36(4):33–347

Erkan G, Radev R (2004) LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479

Eugster M, Leisch F. (2009): "From Spider-Man to HeroArchetypal Analysis in R", *Journal of Statistical Software*, 30(8):1-23.

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

Gong Y, Liu X (2001) Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, ACM, pp 19-25.

Huggins P, Pachter L, Sturmfels B (2007) Toward the Human Genotope. In: *Bulletin of Mathematical Biology*, 69(8) pp 27232735.

Ledeneva Y, René Arnulfo García-Hernández A, Soto R-M, Reyes R-C, Gelbukh A-F (2011) EM Clustering Algorithm for Automatic Text Summarization. In: *Proceedings of Advances in Artificial Intelligence - 10th Mexican International Conference on Artificial Intelligence*, LNCS, Springer, pp 305-315.

Lee J-H, Park S, Ahn CM, Kim D (2009) Automatic generic document summarization based on non-negative matrix factorization. *Inf. Process. Manage.*, 45(1):20–34

Lee C-B, Kim M-S, Park H-R (2003) Automatic Summarization Based on Principal Component Analysis. In: *Proceedings of Progress in Artificial Intelligence*, LNCS, Springer, pp 19-25.

Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: *Text summarization branches out: proceedings of the ACL-04 workshop of ACL 2004*, pp 7481.

Lin C-Y, Hovey E (2003) Automatic evaluation of summaries using n-gram co-occurence statistics. In: *Proceedings of the 2003 conference of the North American chapter of the association for computa-tional linguistics on human language technology*, HLT-NAACL, pp 7178.

Mani I (1991) Automatic summarization, John Benjamins Publishing Company.

Mei J-P, Chen L (2012) SumCR: A new subtopic-based extractive approach for text summarization. *Knowl. Inf. Syst.*, 31(3):527–545

Mei Q, Guo J, Radev D-R (2010) DivRank: the interplay of prestige and diversity in information networks. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, ACM, pp 1009-1018.

Mihalcea R, Tarau P (2004) TextRank: Bringing Order into Text. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP, ACL, pp 404-411.

Mørup M, Hansen L-K (2012) Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54–63.

Page L, Brin S, Motwani R, Winograd T (1998) The PageRank citation ranking: bringing order to the web. *Stanford University*

Porzio G-C, Ragozini G, Vistocco D. (2008) On the use of archetypes as benchmarks. *Applied Stochastic Models in Business and Industry*, 24(5):419-437.

Baeza-Yates R., Berthier R, (1999) Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Richardson M, Domingos P (2001) The Intelligent surfer: Probabilistic Combination of Link and Content Information in PageRank. In: *Proceedings of the Advances in Neural Information Processing Systems 14*, NIPS, MIT Press, pp 1441-1448.

Steinberger J, Jezek K (2004) Text Summarization and Singular Value Decomposition. In: *Proceedings of Advances in Information Systems*, ADVIS, Springer, pp 245-254.

Wang D, Zhu S, Li T, Chi Y, Gong Y (2011) Integrating Document Clustering and Multidocument Summarization. *TKDD*, 5(3):14

Wang D, Li T, Zhu S and Ding C (2008) Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR08)*, pp 307-314.

Wei F, Li W, Lu Q, He Y (2010) A document-sensitive graph model for multi-document summarization. *Knowl. Inf. Syst.*, 22(2):245–259

Yeh JY, Ke HR, Yang WP and Meng IH (2005) Text summarization using a trainable summarizer and latent semantic analysis. *Inf. Process. Manage.*, 41(1):75–95

Zhu X, Goldberg A-B, Gael J-V, Andrzejewski D (2010) Improving Diversity in Ranking using Absorbing Random Walks. In: *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL, pp 97-104.