

*Graph-based models for multi-document  
summarization*

A DISSERTATION PRESENTED  
BY

Ercan Canhasi

TO  
THE FACULTY OF COMPUTER AND INFORMATION SCIENCE  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN THE SUBJECT OF  
COMPUTER AND INFORMATION SCIENCE



Ljubljana, 2014



## APPROVAL

*I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.*

— Ercan Canhasi —

April 2014

THE SUBMISSION HAS BEEN APPROVED BY

Dr. Igor Kononenko

*Professor of Computer and Information Science*

ADVISOR AND EXAMINER

Dr. Marko Robnik Šikonja

*Associate Professor of Computer and Information Science*

EXAMINER

Dr. Dunja Mladenčić

*Associate Professor of Computer and Information Science*

EXTERNAL EXAMINER

J. Stefan Institute



## PREVIOUS PUBLICATION

I hereby declare that the research reported herein was previously published/submitted for publication in peer reviewed journals or publicly presented at the following occasions:

- [1] Ercan Canhasi and I. Kononenko. Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems with Applications*, 41(2):535–543. 2014.
- [2] Ercan Canhasi and I. Kononenko. Multi-document summarization via Archetypal Analysis of the content-graph joint model. *Knowledge and Information Systems*, 1–22, 2013. doi: [10.1007/s10115-013-0689-8](https://doi.org/10.1007/s10115-013-0689-8)
- [3] Ercan Canhasi and I. Kononenko. Semantic role frames graph-based multi-document summarization *Proc. of Conference on Data Mining and Data Warehouses. SiKDD* 2011

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Ljubljana.



## POVZETEK

Glavna tema te disertacije je samodejno povzemanje skupin besedil (angl. multi-document summarization, MDS). Predstavimo rezultate poskusov pri nalogah splošnega povzemanja, povzemanja s poižvedbami, posodabljanja povzetka, in primerjalnega povzemanja skupin besedil. Opišemo obstoječe rešitve in naše izboljšave nekaterih pomembnih delov sistema za povzemanje, vključno z modeliranjem besedila z uporabo grafov in izbiro stavkov z uporabo analize z arhetipi. Osrednji prispevek dela je nova metoda za povzemanje besedil,

Analiza z arhetipi (AA) je obetavna metoda nenadzorovanega strojnega učenja, ki združuje prednosti združevanja v skupine in fleksibilnost matrične faktorizacije. Če pri nalogi splošnega povzemanja besedil množico besedil predstavimo z grafom, bodo pozitivno in/ali negativno najbolj pomembni stavki vrednosti na robu množice podatkov. Za izračun teh ekstremnih vrednosti, splošnih ali uteženih arhetipov, smo se odločili za uporabo AA in utežene AA.

Vsak stavek v množici podatkov modeliramo kot mešanico arhetipnih stavkov. Pri izbiri arhetipov se omejimo na redko posejane mešanice - konveksne kombinacije izvirnih stavkov. Ker AA že vključuje mehko združevanje v skupine in rangiranje, jo uporabimo za sočasno združevanje in rangiranje stavkov. Pomemben argument v prid uporabe AA v MDS je tudi dejstvo, da AA izbira različne (arhetipne) stavke, medtem ko druge metode faktorizacije izbirajo prototipne, karakteristične ali celo osnovne stavke. Torej, z uporabo AA zagotovimo bolj pestre povzetke.

Prispevek našega dela so tudi novi pristopi k modeliranju za povzemanje besedil, ki temeljijo na grafih. Raziskali smo učinke uporabe modeliranja z grafom vsebine ali večelementnim grafom pri nalogah splošnega povzemanja besedil in povzemanja s poižvedbami, neodvisno od jezika in tematike besedil. Predlagamo tudi novo različico AA - uteženo hierarhično AA - in raziščemo njeno uporabnost pri štirih najbolj pogostih

nalogah pri povzemanju besedil: splošnem povzemanju, povzemanju s poizvedbami, posodabljanju povzetka in primerjalnem povzemanju. Učinkovitost in uspešnost predlaganih metod na različnih nalogah preverimo s poskusi na znanih množicah podatkov za povzemanje besedil (DUCo4-07, TACo8).

*Ključne besede:* povzemanje skupin besedil, analiza z arhetipi, utežena analiza z arhetipi, utežena hierarhična analiza za arhetipi, matrična dekompozicija, graf vsebine, večelementni graf, povzemanje s poizvedbami, posodabljanje povzetka, primerjalno povzemanje.



## ABSTRACT

This thesis is about automatic document summarization, with experimental results on general, query, update and comparative multi-document summarization (MDS). We describe prior work and our own improvements on some important aspects of a summarization system, including text modeling by means of a graph and sentence selection via archetypal analysis. The centerpiece of this work is a novel method for summarization that we call "Archetypal Analysis Summarization".

Archetypal Analysis (AA) is a promising unsupervised learning tool able to completely assemble the advantages of clustering and the flexibility of matrix factorization. We propose a novel AA based summarization method based on following observations. In generic document summarization, given a graph representation of a set of documents, positively and/or negatively salient sentences are values on the data set boundary. To compute these extreme values, general or weighted archetypes, we choose to use archetypal analysis and weighted archetypal analysis, respectively. 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 and probabilistic ranking, we suggest considering 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. Our research contributes by presenting some new modeling approaches based on graph notation which facilitate the text summarization task. We investigate the impact of using the content-graph and multi-element graph model for language- and domain-independent extractive multi-document generic and query focused summarization. We also propose the novel version of AA, the weighted Hier-

archical Archetypal Analysis. We consider the use of it for four best-known summarization tasks, including generic, query-focused, update, and comparative summarization. Experiments on summarization data sets (DUCo4-07, TACo8) are conducted to demonstrate the efficiency and effectiveness of our framework for all four kinds of the multi-document summarization task.

*Key words:* Multi-document summarization, Archetypal Analysis, weighted Archetypal Analysis, weighted Hierarchical Archetypal Analysis, Matrix decomposition, Content-graph joint model, Multi-element graph, Query-focused summarization, Update summarization, Comparative summarization.

## ACKNOWLEDGEMENTS

*Prof. Dr. Igor Kononenko has been an exemplary supervisor who always has time for me. Without his support, guidance and interest this dissertation would not have been possible. Communicating with him (not only about work) has always been a pleasure. I would like to thank Igor: he was role model in many professional and human aspects and had prepared me for academic life.*

*Next I would like to thank my colleagues from laboratory for cognitive modeling for a friendly working atmosphere and for their constant willingness to help me. Dr. Darko Pevec and Assist. Prof. Dr. Erik Štrumbelj are very special friends, I thank them for their readiness to help me whenever I needed.*

*Thanks also to my committee members for reading my thesis and providing valuable feedback. Special thanks go to Assoc. Prof. Dr. Dunja Mladenić and Assoc. Prof. Dr. Marko Robnik Šikonja, who greatly influenced this work by providing many detailed and challenging comments.*

*It is impossible to overemphasize the support of my family, especially my beloved wife Riana for giving me generous support and understanding in every aspect of my life. I would also like to extend my gratitude to my parents, in particular my mother, for all of the long-time support they have given to me, visible and invisible. Our baby Rumi has a distinguished complementary role in my well-being.*

— Ercan Canhasi, Ljubljana, April 2014.



# CONTENTS

<i>Povzetek</i>	<i>i</i>
<i>Abstract</i>	<i>iii</i>
<i>Acknowledgements</i>	<i>v</i>
<b>1</b> <i>Introduction</i>	<b>1</b>
1.1 Research Topics . . . . .	3
1.2 Contributions . . . . .	4
1.3 Organization of the Thesis . . . . .	6
1.4 Notation . . . . .	7
<b>2</b> <i>Background and Related Work</i>	<b>9</b>
2.1 Graph theory and summarization . . . . .	10
2.1.1 Terminology, notations and representations . . . . .	10
2.1.2 A simple illustrative example . . . . .	13
2.2 Multi-Document Summarization . . . . .	14
2.2.1 General summarization . . . . .	14
2.2.2 Query-focused summarization . . . . .	19
2.2.3 Update and comparative summarization . . . . .	21
2.3 Archetypal Analysis . . . . .	22
2.3.1 Archetypal Analysis . . . . .	22
2.3.2 Weighted AA . . . . .	22

3	<i>Archetypal Analysis of the content-graph joint model for generic multi-document summarization</i>	27
3.1	Introduction . . . . .	28
3.2	AASum - Archetypal Analysis based document Summarization . . . . .	29
3.2.1	Archetypal Analysis . . . . .	29
3.2.2	MDS problem statement and corpus modeling . . . . .	31
3.2.3	Generic document summarization by AASum . . . . .	36
3.2.4	An illustrative example . . . . .	38
3.2.5	Discussions and Relations . . . . .	41
3.3	Experiments . . . . .	43
3.3.1	Experimental data and evaluation metric . . . . .	43
3.3.2	Input matrix selection and it's impact on summarization . . . . .	44
3.3.3	Impact of the archetype algorithm's initialization on summarization performance and on the speed of the convergence . . . . .	45
3.3.4	Impact of the number of archetypes . . . . .	46
3.3.5	Comparison with related methods . . . . .	48
3.4	Conclusion and future work . . . . .	52
4	<i>Weighted Archetypal Analysis of the multi-element graph for query-focused multi-document summarization</i>	55
4.1	Introduction . . . . .	56
4.2	Weighted Archetypal Analysis . . . . .	57
4.3	Multi-element Graph Model . . . . .	58
4.4	Query-focused document summarization by wAASum . . . . .	61
4.4.1	wAASum . . . . .	61
4.4.2	An illustrative example . . . . .	65
4.5	Experiments . . . . .	66
4.5.1	Experimental data and evaluation metric . . . . .	67
4.5.2	Multi-element graph modeling and it's impact on summarization . . . . .	68
4.5.3	Comparison with related methods . . . . .	70
4.5.4	Impact of the number of archetypes . . . . .	70
4.6	Conclusion and future work . . . . .	71

5	<i>Weighted Hierarchical Archetypal Analysis based generic multi-document summarization framework</i>	73
5.1	Introduction	74
5.2	weighted Hierarchical Archetypal Analysis	75
5.2.1	An illustrative example of wHAA	75
5.2.2	General outline	77
5.3	Summarization method using wHAA	78
5.3.1	Why weighted Hierarchical Archetype Analysis	78
5.3.2	wHAASum algorithm	78
5.3.3	An illustrative example	80
5.4	The summarization framework	83
5.4.1	General summarization	85
5.4.2	Query-focused summarization	85
5.4.3	Update summarization	86
5.4.4	Comparative summarization	87
5.5	Experiments	88
5.5.1	Generic summarization	89
5.5.2	Query-focused summarization	92
5.5.3	Update summarization	93
5.5.4	Comparative summarization	95
5.6	Conclusion and future work	96
6	<i>Final Discussion</i>	97
6.1	Complexity Analysis	98
6.1.1	Preprocessing	98
6.1.2	Archetypal Analysis	98
6.1.3	Archetypal analysis based document summarization	99
6.2	Limitations	100
6.2.1	Evaluation	100
6.2.2	Guided Summarization	102
6.2.3	The length of the selected sentences and its impact on summarization quality	106
6.2.4	Evaluation of the AA based summarization methods on some newer data sets	108

7	<i>Conclusion and Future Work</i>	<i>III</i>
7.1	Conclusion . . . . .	<i>112</i>
7.2	Future Work . . . . .	<i>113</i>
A	<i>Razširjeni povzetek</i>	<i>115</i>
A.1	Uvod . . . . .	<i>116</i>
A.2	Raziskovalne teme . . . . .	<i>117</i>
A.3	Analiza z arhetipi . . . . .	<i>118</i>
A.3.1	Splošno povzemanje besedil z uporabo analize z arhetipi . . . . .	<i>118</i>
A.4	Poskusi . . . . .	<i>122</i>
A.4.1	Primerjava s sorodnimi metodami. . . . .	<i>122</i>
A.5	Zaključek . . . . .	<i>125</i>
A.5.1	Prispevki znanosti . . . . .	<i>125</i>
A.5.2	Nadaljnje delo . . . . .	<i>127</i>
	<i>Bibliography</i>	<i>129</i>



*Introduction*

The main objective of this thesis is the development of a new text summarization method that would take advantage of graph modeling flexibility and archetypal analysis efficiency.

Half a century has passed since the publication of Luhn's pioneering paper on automatic summarization [1]. All along this time the pragmatic need for automatic summarization has become more and more important and many papers have been published on the topic. The World Wide Web consist of billions of documents containing information, mostly textual, and still growing exponentially. These facts have triggered interest in the development of automatic document summarization systems. First, such systems were designed to take a single article or a cluster of news articles, and produced a brief and natural summary of the most important information. These systems, immature as they are, have been already found very useful by human and other automatic applications and interfaces. Recently many novel summarization tasks and applications has been developed and reported in literature. Extractive summaries (extracts) are generated by attaching a few sentences taken exactly as they occur in the document(s) being summarized. Abstractive summaries (abstracts) are generated to disclose the main essence of the original document(s). Even though abstracts may rephrase or even use the original sentences, they are generally expressed in the words of the author. Single document summarization was the very first task treated by early summarization works, where systems produced a summary of one document. As research advanced, the multi-document summarization as a new type of summarization task emerged. Multi-document summarization (MDS) was motivated by use cases on the web. Given the large amount of redundant textual data on the web, MDS can be very useful tool when used to produce a brief summary of many documents on the same topic or the same event. In the first utilized online summarization systems, the MDS was applied to clusters of news articles on the same event to produce online browsing pages [2]. Summaries can also be identified by their gist. A summary that provides the reader with subject of the original documents is often called an indicative summary. A summary that can be read in place of the document is called an informative summary. An informative summary will include facts that are reported in the input document(s), while an indicative summary may provide characteristics such as length, writing style, etc.

In this thesis we take the less common approach to summarization problem, i.e we treat the extractive summarization task (1) through modeling text by means of

similarity graphs and (2) by selecting sentences via Archetypal Analysis (AA). We model text in many different ways, as a similarity graph, as a content graph, as a content-graph joint model, and as a multi-element graph. All those modeling methods are detailed in the following chapters. For sentence selection we take less common approach of treating the problem of sentence selection as the mixture of matrix decomposition and low rank approximation approaches. In other words, we formalize the sentence selection as the archetypal analysis problem. This approach has many useful properties, which later are described in details.


This dissertation contributes to text mining research in general, while specifically contributes to increasing research interest in document summarization sub-field with application to few different summarization tasks.

### *1.1 Research Topics*

We develop new document summarization methods based on graph models. We investigate four different document summarization problems, namely general, query-focused, update and comparative summarization.

*General summarization.* Up to date many works have investigated the problem of general summarization. It is based on some basic assumptions about the aim of the summary. Given that no assumptions are made about the type of the document(s) that need to be summarized, the content of the input alone is enough to decide on the significance of information. Additionally, the strongest and the most general assumption made in general summarization is that the summary should help the reader easily find out what the documents are about. The last assumption makes the problem of general summarization very hard and it is also the main reason why the other more specific summarization tasks, such as query oriented summarization and guided summarization, have been lately proposed. In the first part of this thesis, we consider the task of general multi-document summarization. To this end, we propose a novel archetypal analysis based extractive summarization model and experiment with it on some standard test sets while measuring how well the model is able to summarize the given documents. We investigate whether archetypal analysis can be used for general summarization. Can archetypal analysis be used as the sentence selection method in a graph based summarization model? Can this model be used with content and graph models jointly? Is this approach useful practically? Is this approach efficient?

*Query-focused summarization.* On the other hand, the query focused summarization



can be used to summarize exclusively the information that is correlated to a specific user-defined query. For instance, given a query and a set of relative documents retrieved by the search engine, a summary of each document can simplify the process of answering the information need expressed by the query. Yet another useful query focused application is producing snippets for search engines. An automatic extractive summarization system in order to produce a useful query focused summary needs to summarize well the given documents in various ways while purposely being biased toward the given query. In the third chapter, we present our novel weighted archetypal analysis based query oriented summarization system. We investigate whether weighted AA can be used as a query focused sentence selection method. We also examine our method in combination with few different graph modeling methods.

*Update and comparative summarization.* The update summarization task requires summarizing a set of documents under the assumption that the reader has already read and summarized the first set of documents as the main summary. For generating the update summary, some clever solutions are required to capture the temporally evolving information and avoid the redundant information which has already been covered by the main summary. The comparative document summarization was first proposed by [3] to summarize differences between comparable document groups. In the fourth chapter, we propose a new framework for MDS using the weighted hierarchical Archetypal Analysis (wHAASum). Many known summarization tasks, including generic, query-focused, update, and comparative summarization, can be modeled as different versions acquired from the proposed framework. We investigate whether the novel framework is suitable for many well known summarization tasks. We also investigate its effectiveness.

## 1.2 Contributions

The reported work contributes new summarization methods (contributions 1, 2 and 3), proposes some new methods for input text modeling by means of graphs in the domain of document summarization (contributions 4,5) and finally advances the state-of-the-art in four known document summarization tasks, including general, query-focused, update and comparative summarization.

1. *Archetypal Analysis Summarization (AASum) method.* The AASum method introduces an archetypal analysis based approach for identifying a subset of repre-

sentative and diverse sentences from a document collection.

- We propose a novel use of the archetypal analysis(AA) method to guide a search for representative sentences, by measuring the distance of sentences from some positively and/or negatively outstanding archetypal sentences identified by AA.
- We develop an efficient algorithm for summary sentence selection based on AA. Hereafter, by the efficiency of our summarization methods we mean the high scores of the summary evaluation, that are detailed in experimental work.
- We empirically demonstrate the efficiency of the proposed summarization method for general summarization task.

2. *weighted Archetypal Analysis Summarization (wAASum) method.* The wAASum method proposes a weighted archetypal analysis based approach for extracting a subset of representative and diverse sentences from a document collection given the user defined query.

- We propose a new use of the weighted archetypal analysis (wAA) method to regulate the search for representative sentences, by measuring the distance of sentences from some weighted positively and/or negatively outstanding weighted archetypal sentences identified by wAA.
- We develop an efficient algorithm for query-focused summary sentence selection based on wAA.
- We empirically demonstrate the effectiveness of the proposed summarization method for query-focused summarization task.

3. *weighted Hierarchical Archetypal Analysis Summarization (wHAASum) method.* The wHAASum method proposes a weighted hierarchical version of the archetypal analysis based approach for summary extraction.

- We present a novel version of archetypal analysis problem. To the best of our knowledge, the problem of hierarchical wAA has not been proposed or studied before.

- We propose a new use of the weighted hierarchical archetypal analysis (wHAA) method to regulate the search for representative sentences, by measuring the distance of sentences from the “best of the best” sentences identified by wHAA.
  - We develop an efficient framework for all known summarization tasks, including general, query-focused, update and comparative summarization.
  - We empirically demonstrate the effectiveness of the proposed summarization framework.
4. *Content-graph joint model.*
- The content-graph joint model is a novel way for input document modeling in summarization.
  - It provides a methodical way of combining information from both the terms and sentence similarity connection structure present in the corpus.
  - We show that AASum performs much better in terms of effectiveness when the content graph joint model is used.
5. *Multi-element graph model.*
- We introduce the modeling of input documents and query information as a multi-element graph model.
  - We show that wAASum performs well in terms of effectiveness when the multi-element graph model is used.

### 1.3 Organization of the Thesis

The remainder of the thesis is structured as follows.

*Chapter 2* first presents the background of document summarization in general and the graph based summarization specifically. We also review prior work which considers the graph based summarization systems where sentence selection is done by algebraic methods (even though the works similar to different contributions of this thesis are also discussed in later chapters when it is needed). This chapter places our work in a wider context of graph based models of text modeling and algebraic approaches to

sentence selection, and covers related materials for reader not familiar with the specific field.

*Chapter 3* describes our first attempt on treating the most difficult and the most general summarization problem, i.e general MDS. We present our novel extractive summarization method based on Archetypal Analysis. This method is the corner-stone for all other methods described later in this work.

*Chapter 4* presents the use of weighted Archetypal Analysis in query oriented summarization. The novel method proposes a weighted archetypal analysis based approach for extracting a subset of representative and diverse sentences from a document collection given the user defined query.

*Chapter 5* reports on an application of a novel version of archetypal analysis, namely weighted Hierarchical Archetypal Analysis for extractive summarization method for many known summarization tasks.

*Chapter 6* presents the final discussion including the complexity analysis, some of the limitations in evaluation and guided summarization, and our proposals for treating them.

*Chapter 7* concludes the thesis with a condense summary and a direction to future work.

## 1.4 Notation

This dissertation assumes that the reader is familiar with the fundamentals of the linear algebra. Furthermore, it is assumed that the reader is familiar with Euclidean vector spaces and corresponding vector operations. This section is included solely to introduce the notation that is employed, as needed, throughout the dissertation.

- $n$  - number of sentences in a document set (observations),
- $t$  - total number of sentences in all document sets,
- $y$  - number of documents in a documents set,
- $m$  - number of terms (variables),
- $z$  - number of (required) archetypes,
- $l$  - number of (required) sentences in a summary,

- $N$  - number of vertices in a graph,
- $T$  - total number of documents in all document sets,
- $k$  - number of levels (of the hierarchy of archetypes),
- $s_i, s_j$  - sentences,
- $q$  - query,
- $sim_{norm}(s_i, s_j)$  - normalized similarity,
- $D$  - document set,
- $SM$  - generic summary,
- $SM_i$  - summary for document set  $D_i$ ,
- $G_s$  - general summary,
- $U_s$  - update summary,
- $C_s$  - compare summary,
- $[A]_{n \times n}$  - sentence similarity matrix,
- $[T]_{m \times n}$  - term-sentence matrix,
- $[TA]^T = [X]_{n \times m}$  - content graph joint model,
- $[W]_{n \times n}$  - weight diagonal matrix,
- $[Y]_{m \times z}$  - the matrix of  $z$  archetypes where rows represent variables (terms),
- $[C]_{n \times z}$  - decomposition matrix used by AA to get  $X \approx SY^T = S(X^T C)^T = SC^T X$ ,
- $[S]_{n \times z}$  - decomposition matrix used by AA to get  $X \approx SY^T = S(X^T C)^T = SC^T X$ ,
- $\|\cdot\|^2$  - denotes the Euclidean matrix norm,
- $\odot$  - denotes the Hadamard matrix product
- $\otimes$  - denotes the inner matrix product.



## *Background and Related Work*

We begin with a brief introduction to graph theory and its connection to document summarization. In Subsection 2.1.2 a simple illustrative example of graph-based summarization system is given. In Section 2.2, an introduction to Multi-document summarization (MDS) and detailed description of the state-of-the art of algebraic summarization systems are presented. Then, in Section 2.3, an introduction to Archetypal Analysis (AA) and its weighted version with the relevant work regarding them are given. Finally, we conclude the chapter exposing the description of the research problem of this dissertation.

### 2.1 *Graph theory and summarization*

Language entities such as words, phrases, and complete sentences, originating from a meaningful text, are related with various relationships. Those connections contribute to the total context and support the structure and text unity. Semantic networks have been first proposed in the early days of artificial intelligence as descriptions that make possible the simultaneous storage of the content (i.e language units) and structure (i.e. interconnecting links). Multiple kind of deduction and reasoning processes that simulate the human mind [4] can be modeled with this powerful tool. Graphs are the natural equivalents to the mathematical structures that originate from these descriptions. In them the text units are represented as vertices and their interconnecting relationships as the edges.

Graphs can be used in modeling many natural language processing applications. For a very detailed review and the comprehensive description of the use of graph-based algorithms for natural language processing and information retrieval see [5]. Graphs as very descriptive data structures are known for their ability to readily encode the semantic content and syntactic structure of a meaningful text. For instance in Figure 2.1 are given some examples of the mentioned graph representations where (a) (adopted from [6]) represents a structure graph of a text by encoding similarity relationships among textual units; (b) represents a graph with eight nodes modeling a word-meaning problem; and (c) illustrates a sample graph built to extract semantic classes.

#### 2.1.1 *Terminology, notations and representations*

Graph is a data structure composed of a set of nodes connected by a set of edges. Graphs can be used to model relationships among the objects in a collections and

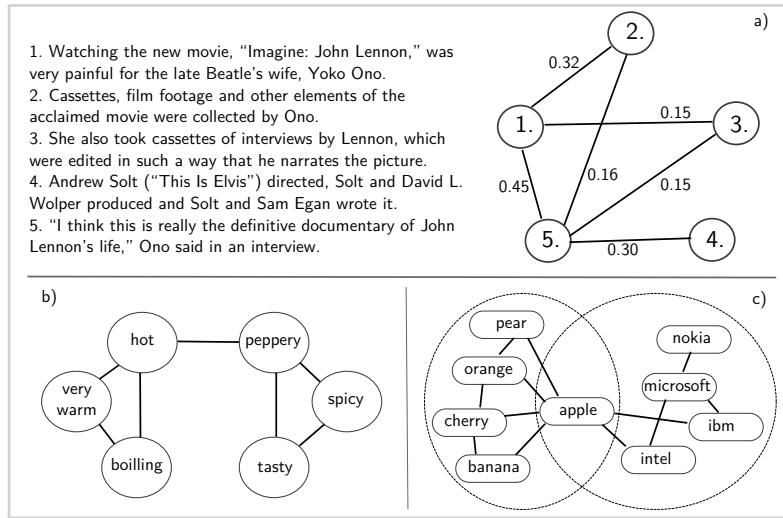


Figure 2.1

Examples of the graph representations of various textual units and their adjacent connections.

they are traditionally studied in a study area of mathematics known as a graph theory. Figure 2.2 presents the celebrated seven bridges in Königsberg that forced the creation of graph theory. The problem is to pass over all bridges, but only once, without crossing any bridge twice or more. Back in 1741, Euler proved that it is not possible to do so.

Formally, a graph is defined as a set  $G = (V, E)$ , where  $V$  is a collection of nodes  $V = \{V_i, i = 1, \dots, N\}$  and  $E$  is a collection of edges over  $V$ ,  $E_{ij} = \{(V_i, V_j), V_i \in V, V_j \in V\}$ . Graphs can be either directed or undirected, depending on whether a direction of travel is defined over the edges. In a directed graph (ordigraph), an edge  $E_{ij}$  can be traversed from  $V_i$  to  $V_j$  but not in the other direction;  $V_i$  is called the tail of the edge and  $V_j$  is called the head. In an undirected graph, edges can be traversed in both directions. Figure 2.3(a) is an example of an undirected-graph with four nodes connected by four edges. Figure 2.3(b) is a similar graph but with directed edges. In the figures, 1, 2, 3, 4 are nodes, and (1, 2), (2, 3), (2, 4), and (3, 4) are edges that connect them.

If two nodes,  $V_i$  and  $V_j$ , are connected by an edge, they are said to be adjacent. The edge  $E_{ij}$  connecting them is said to be incident on the two nodes  $V_i$  and  $V_j$ . In a directed graph, because an edge implies a direction of traversal, the tail of an edge is

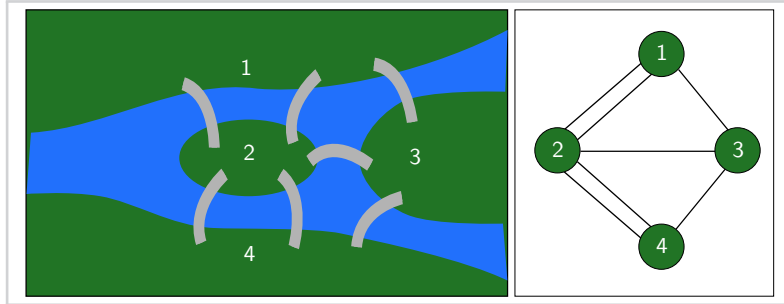


Figure 2.2

The seven bridges in Königsberg (schematic and graph).

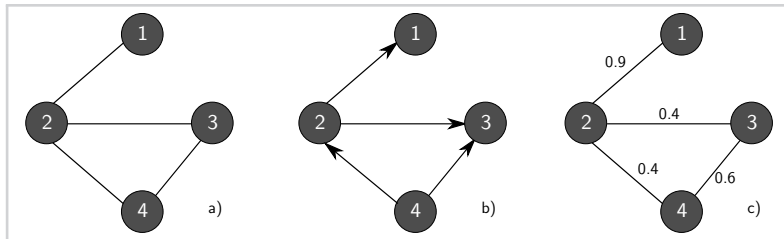


Figure 2.3

A sample graph: (a) undirected, (b) directed, and (c) weighted undirected.

said to be adjacent to the head of the edge; conversely, the head of the edge is said to be adjacent from the tail of the edge. A graph in which every two nodes are adjacent to one another is called a complete graph. By default, edges in a graph are unweighted; thus, the corresponding graphs are said to be unweighted. When a weight is assigned to the edges in the graph, as in Figure 2.3(c), the graphs are said to be weighted. A vector  $V$  is a  $1 \times N$  array of objects (such as numbers), whereas a matrix  $M$  is an  $N \times N$  array. Vectors can be used to represent the coordinates of a point. Matrices often are used to represent ordered collections of vectors. For example, a set of four vectors, all having the same length of five elements, is represented collectively as a  $4 \times 5$  matrix.

Figure 2.4

Matrix representation for the graph in Figure 2.3(b).

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & 0.9 & 0 & 0 \\ 0.9 & 0 & 0.4 & 0.4 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0.4 & 0.6 & 0 \end{bmatrix}$$

Figure 2.5

Matrix representation for the graph in Figure 2.3(c).

Graphs and matrices often are used interchangeably to represent relational data. Figure 2.4 shows the matrix representation for the graph  $G$  shown in Figure 2.3(b). It has four nodes (1 through 4) and four edges (2, 1), (2, 3), (2, 4), (4, 3).  $G$  is a directed graph because not all relationships are symmetric.  $G$  can be represented as a matrix, in which the rows and columns correspond, in order, to the nodes 1 through 4, and a value of 1 indicates the presence of a directed edge between the corresponding nodes. Undirected graphs are represented in a similar way, but an edge is represented redundantly by using two cells. For instance, the edge (2, 3) is represented using a value of 1 stored in the cells corresponding to the (directed) edges (2, 3) and (3, 2). Alternatively, undirected graphs also can be represented using diagonal matrices to avoid redundancy. Finally, weighted graphs can be represented using a similar matrix structure, with the values of the cells corresponding to the weight of the edges. For example, the matrix in Figure 2.5 is the representation of the weighted undirected-graph structure from Figure 2.3(c).

The matrix representation of a graph can enable faster computation of graph properties. However, this representation is not always applicable because it tends to grow very quickly in size. For instance, whereas a graph of ten elements can be represented using a matrix of  $10 \times 10 = 100$  elements, a graph of a thousand nodes requires a matrix of a million cells, and so on. Storing only the non-zero values can considerably minimize the memory requirements. Depending on the frequency and distribution of the non-zero values, various data structures can be used and archive significant savings in memory.

### 2.1.2 A simple illustrative example

The idea of graph-based summarization has for the first time attract a wider interest in document summarization community with introduction of the concept of lexical centrality [6–8]. In a graph of lexically and/or semantically related sentences the lexical centrality is a value of significance of the nodes. The most common way of calculat-

ing these values is by execution of a random walk on this graph and the consecutive selection of the most frequently visited nodes as the summary of the input graph.

Additionally, to avoid nodes with duplicate or near duplicate content, the final decision about including a node in the summary also can rely on its maximal marginal relevance. To illustrate, Table 2.1 is an example drawn from [6]: The input consists of eleven sentences from several news stories on related topics, with the matrix of cosine similarities for all sentence pairs shown in Figure 2.6. It is interesting that the cosine matrix could be expanded potentially into an infinite number of graphs for different values of a cosine cutoff.

This is illustrated in the last two subfigures of the Figure 2.6, which show two graphs obtained for two different threshold values. For example, if the threshold is lowered too much, the graph is almost fully connected (Figure 2.6(a)). Conversely, raising the threshold eventually turns the graph into a set of disconnected components (Figure 2.6(b-c)). The random walk is typically performed at a threshold value at which approximately half of the node pairs are connected via edges. For instance, Figure 2.6(a) is the weighted graph built for the text in Table 2.1. The random-walk summarization method can be applied to both single and multi-document summarization because the graph can be built based on information drawn from one or multiple documents. When evaluated on standard datasets from the Document Understanding Conferences, the random-walk summarization method was found to be competitive with other more complex supervised systems. More important, the improvements were consistent across different datasets, covering single-document and multi-document summarization, as well as the summarization of long documents such as books [9].

## 2.2 *Multi-Document Summarization*

Even though there are many ways of presenting the previous work in text summarization, we choose to do it: (1) based on different summarization tasks presented through many years of research in the field, (2) and by strictly persisting in the sub-field of summarization methods known as algebraic summarization methods.

### 2.2.1 *General summarization*

In recent years, algebraic methods, more precisely matrix decomposition approaches have become a key tool for document summarization. Typical approaches used in MDS spread from low rank approximations such as singular value decomposition

Table 2.1

A cluster of 11 related sentences.

ID	Text
$d_1s_1$	Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
$d_2s_1$	Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq rejects cooperating with the United Nations except on the issue of lifting the blockade imposed upon it since the year 1990.
$d_2s_2$	Ramadan told reporters in Baghdad that Iraq cannot deal positively with whoever represents the Security Council unless there was a clear stance on the issue of lifting the blockade off of it.
$d_2s_3$	Baghdad had decided late last October to completely cease cooperating with the inspectors of the United Nations Special Commission (UNSCOM), in charge of disarming Iraq's weapons, and whose work became very limited since the fifth of August, and announced it will not resume its cooperation with the Commission even if it were subjected to a military operation.
$d_3s_1$	The Russian Foreign Minister, Igor Ivanov, warned today, Wednesday, against using force against Iraq, which will destroy, according to him, seven years of difficult diplomatic work and will complicate the regional situation in the area.
$d_3s_2$	Ivanov contended that carrying out air strikes against Iraq, who refuses to cooperate with the United Nations inspectors, "will end the tremendous work achieved by the international group during the past seven years and will complicate the situation in the region."
$d_3s_3$	Nevertheless, Ivanov stressed that Baghdad must resume working with the Special Commission in charge of disarming the Iraqi weapons of mass destruction (UNSCOM).
$d_4s_1$	The Special Representative of the United Nations Secretary-General in Baghdad, Prakash Shah, announced today, Wednesday, after meeting with the Iraqi Deputy Prime Minister Tariq Aziz, that Iraq refuses to back down from its decision to cut off cooperation with the disarmament inspectors.
$d_5s_1$	British Prime Minister Tony Blair said today, Sunday, that the crisis between the international community and Iraq "did not end" and that Britain is still "ready, prepared, and able to strike Iraq."
$d_5s_2$	In a gathering with the press held at the Prime Minister's office, Blair contended that the crisis with Iraq "will not end until Iraq has absolutely and unconditionally respected its commitments" towards the United Nations.
$d_5s_3$	A spokesman for Tony Blair had indicated that the British Prime Minister gave permission to British Air Force Tornado planes stationed in Kuwait to join the aerial bombardment against Iraq.

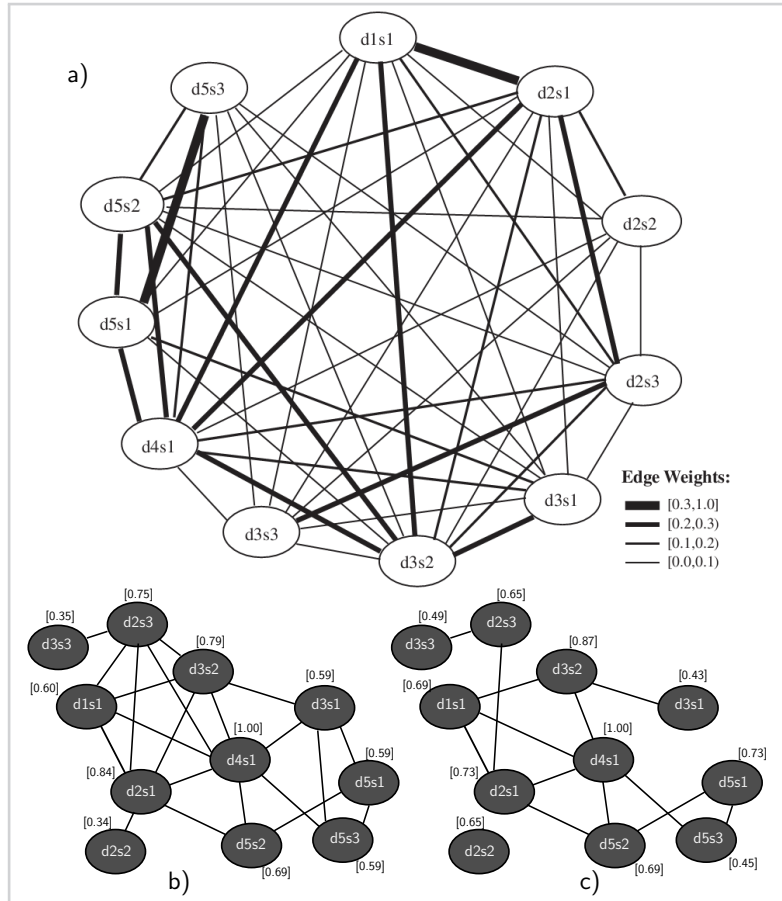




Table 2.2

A cluster of 11 related sentences.

	1	2	3	4	5	6	7	8	9	10	11
1	1.00	0.45	0.02	0.17	0.03	0.22	0.03	0.28	0.06	0.06	0.00
2	0.45	1.00	0.16	0.27	0.03	0.19	0.03	0.21	0.03	0.15	0.00
3	0.02	0.16	1.00	0.03	0.00	0.01	0.03	0.04	0.00	0.01	0.00
4	0.17	0.27	0.03	1.00	0.01	0.16	0.28	0.17	0.00	0.09	0.01
5	0.03	0.03	0.00	0.01	1.00	0.29	0.05	0.15	0.20	0.04	0.18
6	0.22	0.19	0.01	0.16	0.29	1.00	0.05	0.29	0.04	0.20	0.03
7	0.03	0.03	0.03	0.28	0.05	0.05	1.00	0.06	0.00	0.00	0.01
8	0.28	0.21	0.04	0.17	0.15	0.29	0.06	1.00	0.25	0.20	0.17
9	0.06	0.03	0.00	0.00	0.20	0.04	0.00	0.25	1.00	0.26	0.38
10	0.06	0.15	0.01	0.09	0.04	0.20	0.00	0.20	0.26	1.00	0.12
11	0.00	0.00	0.00	0.01	0.18	0.03	0.01	0.17	0.38	0.12	1.00

(SVD) [10, 11], principal component analysis (PCA) [12], latent semantic indexing (LSI/LSA) [13, 14], non-negative matrix factorization (NMF) [15] and symmetric-NMF [16] to soft clustering approaches such as fuzzy K-medoids [17] and the EM-algorithm for clustering [18] and hard assignment clustering methods such as K-means [16]. Graph based methods can also be categorized as decomposition methods as they are based on eigen decomposition which is closely related to the SVD.

Graph-based methods like LexRank [6] and TextRank [8] 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 [19] that has been successfully applied to rank Web pages in the Web graph. The recently proposed document-sensitive graph model [20] that emphasizes the influence of a global document set information on the local sentence evaluation, is shown to perform better than other graph models for multi-document summarization task where MDS is modeled as a single combined document summarization. Although those methods have shown to be successful in covering relevance by calculating the principal or dominant eigenvector, they suffer from some fundamental

limitations such as the lack of diversity in produced summaries [21, 22], and topic drift handling [23]. As these algorithms tend to ignore the influence of eigenvectors other than the largest one, the sentences related to topics other than the 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 [21] 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 [22].

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 the term-document matrix into three matrices,  $U$ ,  $D$ , and  $V$ . Starting from the first row of  $V^T$ , 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 [13, 14]. However, LSA has a number of drawbacks, namely its unsatisfactory statistical foundations. The EM-algorithm for clustering is utilized in work by [24] 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 a generic summary possible.

Automatic generic document summarization based on non-negative matrix factorization [15] 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 a superior representation of the subtopics of doc-

uments. The SNMF summarization framework, as an extension of [15], 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 the closeness to the cluster center and the closeness to the sentences in the same cluster, therefore it is incapable of considering both in defining the subtopic-based feature. A fuzzy medoid-based clustering approach, as presented in [17] 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 different subtopics and thus enhances the chance of producing a summary with a wider coverage and less redundancy.

A method, called MCLR (maximum coverage and less redundancy) [25], models multi-document summarization as a quadratic boolean programming problem where the objective function is a weighted combination of the content coverage and redundancy objectives. Another successful constraint-driven document summarization model is presented in [26]. The model in this work is formulated as a quadratic integer programming problem and solved with a discrete binary particle swarm optimization algorithm. A method, called WHM (weighted harmonic mean function), as presented in [27], models multi-document summarization as an optimization problem where the objective function is a weighted linear combination and a weighted harmonic mean of the coverage and redundancy objectives.

In our 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 the content-graph joint model can better describe the relationships between sentences. Experimental results demonstrate the effectiveness of our proposed framework.

### 2.2.2 Query-focused summarization

Recently, algebraic methods, more precisely matrix factorization approaches, have become an important tool for query/topic focused document summarization. The exemplary methods used until now vary from low rank approximations, such as singular value decomposition (SVD) [10], latent semantic indexing (LSI/LSA) [13, 28], non-negative matrix factorization (NMF) [15, 29] and symmetric-NMF [16] to soft

clustering approaches such as fuzzy K-medoids [17] and hard assignment clustering methods such as K-means [16]. The graph based methods can also be categorized as a factorization methods since they are based on eigendecomposition which is closely related to the SVD.

Graph-based methods like LexRank [6] and TextRank [8] 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. Graph-based ranking algorithms were also used in query-focused summarization when it became a popular research topic. For instance, a topic-sensitive version of LexRank is proposed in [7]. It integrates the relevance of a sentence to the query into LexRank to get a biased PageRank ranking. Although this algorithm is proposed for sentence ranking in the context of question-focused sentence retrieval, it can be directly used for sentence ranking in the task of query-focused summarization. The recently proposed document-sensitive graph model [20] 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.

The 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 Q-MDS method using LSA applies the singular value decomposition (SVD) to summarize documents. This method factorizes a term-document matrix into three matrices,  $U$ ,  $D$ , and  $V$ . Starting from the first row of  $V^T$ , the sentence corresponding to the column that has the largest index value with the right singular vector is selected to the next stage [13, 14]. Then a query focus from a topic description is derived to be used for guiding the sentence selection [28]. However, LSA has a number of drawbacks, due to its unsatisfactory statistical foundations.

In [15] the query based summarization method using NMF is proposed. This method is yet another successful algebraic method, which extracts sentences using the cosine similarity between a query and semantic features. This type of methods conduct NMF on the term-sentence matrix to extract sentences with the highest probability in each topic. Intuitively, this method clusters the sentences and chooses the most repre-

sentative 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. In [29] is also proposed a query based summarization method using NMF. This method extracts sentences using the cosine similarity between a query and semantic features.

The SNMF summarization framework for query focused summarization, as an extension by [15], 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 the closeness to the cluster center and the closeness to the sentences in the same cluster, therefore it is incapable of considering both in defining the subtopic-based features.

A fuzzy medoid-based clustering approach, as presented by [17], is an example of soft clustering methods for Q-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 different subtopics and thus enhances the chance of producing a summary with a wider coverage and less redundancy.

### 2.2.3 *Update and comparative summarization*

Update summarization was first presented at the Document Understanding Conference (DUC) 2007. Then it was the main task of the summarization track at the Text Analysis Conference (TAC) 2008. The update summarization task requires summarizing a set of documents under the assumption that the reader has already read and summarized the first set of documents as the main summary. For generating the update summary, some clever solutions are required to capture the temporally evolving information and avoid the redundant information which has already been covered by the main summary. The timestamped graph model [30], motivated by the evolution of citation network, tries to model the temporal aspect of update summarization. A novel graph based sentence ranking algorithm, namely PNR2, for update summarization as presented in [31], is inspired by the intuition that “a sentence receives a positive influence from the sentences that correlate to it in the same collection, whereas a sentence receives a negative influence from the sentences that correlate to it in the different

(perhaps previously read) collection”. In [32] the authors address a novel content selection framework based on evolutionary manifold-ranking and normalized spectral clustering. The proposed evolutionary manifold-ranking aims to capture the temporal characteristics and relay the propagation of information in a dynamic data stream and the user need.

The comparative document summarization is first proposed in [3] to summarize differences between comparable document groups. The authors present a sentence selection strategy modeled by means of conditional entropy, which precisely discriminates the documents in different groups.

### 2.3 Archetypal Analysis

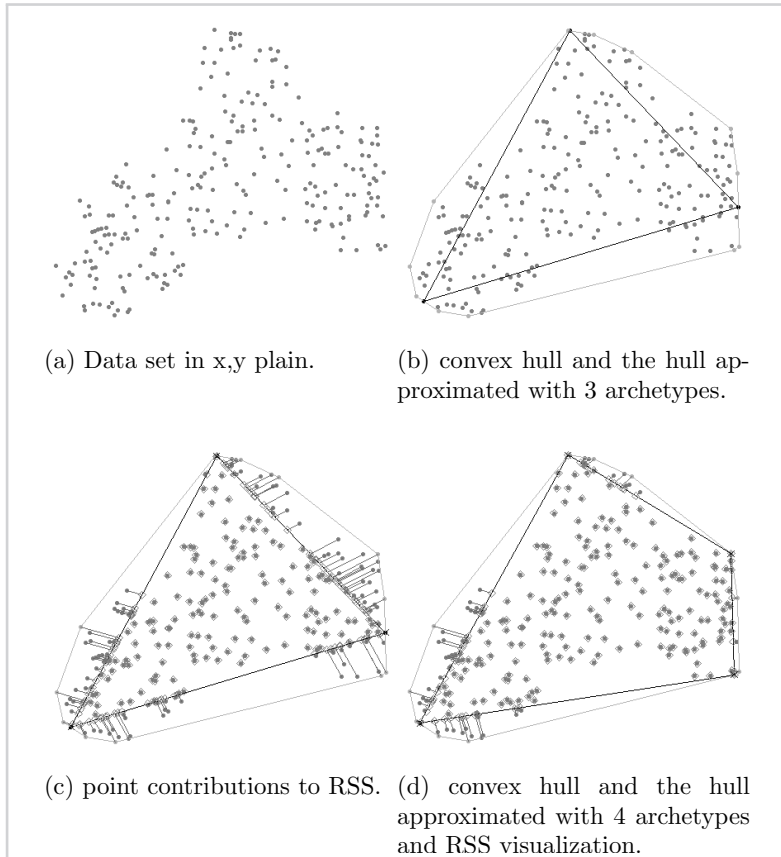
In this subsection we present the related work to the Archetypal Analysis (AA) and the weighted AA.

#### 2.3.1 Archetypal Analysis

Archetypal Analysis as presented by Cutler and Breiman [33] 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. 2.7. 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 different areas, e.g., in economics [34], astrophysics [35] and recently in pattern recognition [36]. 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 [37]. For detailed explanation on numerical issues, stability, computational complexity and implementation of the AA we also refer to [38].

#### 2.3.2 Weighted AA

The weighted version of the Archetypal analysis (wAA) is first introduced in [39] which adapts the original AA algorithm to be a robust M-estimator (i.e. M-estimators are the



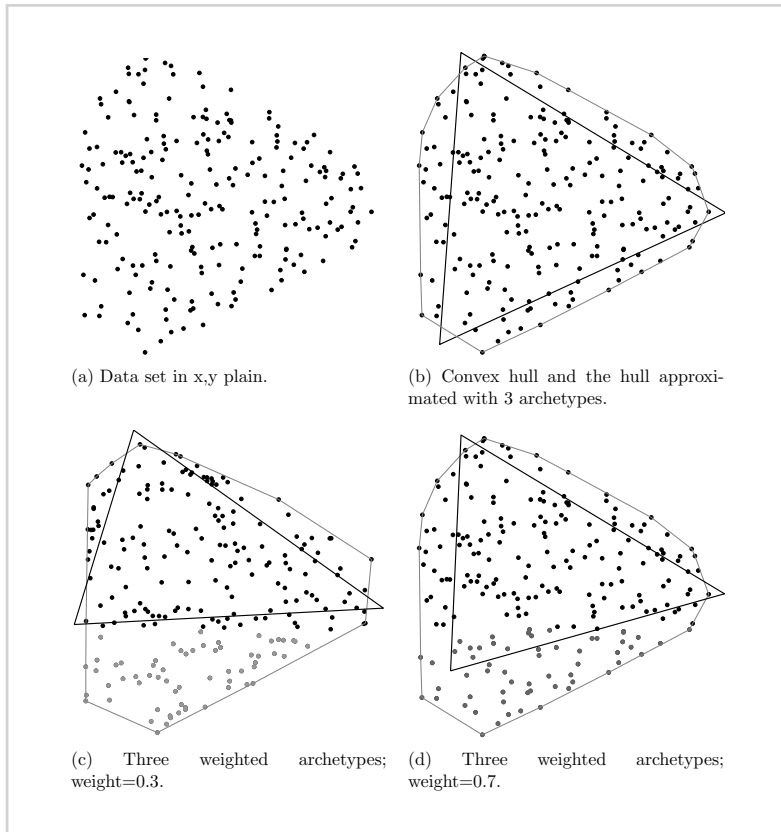
*Figure 2.7*

Archetypal analysis approximates the convex hull of a set of data. Increasing the number  $z$  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.

generalization of estimators, representing the minima of sums of functions of the data). Figure 2.8 compares AA with wAA and also gives the visual gist of wAA.

In this section we mainly presented the background on graph based methods for MDS and archetypal analysis. By using the simple example we showed how a relatively plain method can be successfully used for sentence ranking. In the next section we present our approach of integrating the archetypal analysis in a novel graph based document summarization method.





*Figure 2.8*

Archetypal analysis approximates the convex hull of a set of data (b). Weighted archetypal analysis approximates the weighted convex hull, with respect to points weights (c,d). Here the gray data points weight 0.3 in (c) and 0.7 in (d) while the black data points weight 1 in both. As expected, depending on the points weights the corresponding archetype changes its position inside the weighted data set boundary.



*Archetypal Analysis of the  
content-graph joint model for  
generic multi-document  
summarization*

This chapter is mostly based on our first work published in [40], where the general multi-document summarization problem is treated by using the archetypal analysis of the content-graph joint model.

### 3.1 Introduction

Recently, many generic document summarization methods using matrix factorization techniques have been proposed [10, 12, 13, 15, 16, 16–18]. 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 being able to directly combine the virtues of clustering and the flexibility of matrix factorization with application to the task of MDS is the main investigation objective of this work. In this chapter, we propose a new unsupervised generic document summarization method based on Archetypal Analysis (AA).

The proposed method has the following properties: (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) the extracted sentences can be represented as a 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 the chapter is organized as follows: The details of the proposed summarization approach AASum are presented in Section 3.2, where we give an overview of the new approach, an illustrative example, discussions and relations to similar methods. Section 3.3 shows the evaluation and experimental results. Finally, we conclude in Section 3.4.

### 3.2 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.

#### 3.2.1 Archetypal Analysis

Consider an  $n \times m$  matrix  $X$  representing a multivariate data set with  $n$  observations and  $m$  variables. For given  $z \ll n$  the archetypal problem is to decompose a given matrix  $X$  into stochastic matrices  $S \in \mathbb{R}^{n \times z}$  and  $C \in \mathbb{R}^{n \times z}$  as shown by Eq. (3.1)

$$X \approx SC^T X \quad (3.1)$$

More exactly, the archetypal problem is to find two matrices  $C$  and  $S$  which minimize the residual sum of squares

$$\begin{aligned} \text{RSS}(k) &= \|X - SY^T\|^2 \text{ with } Y = X^T C \\ \text{s.t. } &\sum_{j=1}^z S_{ij} = 1, S_{ij} \geq 0; \sum_{i=1}^m C_{ij} = 1, C_{ij} \geq 0 \end{aligned} \quad (3.2)$$

The constraint  $\sum_{i=1}^z S_{ij} = 1$  together with  $S_{ij} \geq 0$  enforces the feature matrix  $Y$  to be a convex combination (i.e., weighted average) of the archetypes while the constraints  $\sum_{i=1}^m C_{ij} = 1$  and  $C_{ij} \geq 0$ , require that each archetype is a meaningful combination of data points.  $\|\cdot\|^2$  denotes the Euclidean matrix norm.

The description of archetypal analysis given by Eq. (3.2) defines the foundation of the estimation algorithm first presented in [33]. It alternates between finding the best  $S$  for given archetypes  $Y$  and finding the best archetypes  $Y$  for given  $C$ ; 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  $z$ :

1. Pre-processing: scale data.
2. Initialization: initialize  $C$  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  $S$  for the given set of archetypes  $Y$ , i.e. solve  $n$  convex least squares problems, where  $i = 1, \dots, n$

$$\min_{S_i} = \frac{1}{2} \|X_i - Y S_i\|^2 \text{ s.t. } \sum_{j=1}^z S_{ij} = 1, S_{ij} \geq 0.$$

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

$$\min_{C_j} = \frac{1}{2} \|\hat{Y}_j - X C_j\|^2 \text{ s.t. } \sum_{i=1}^m C_{ij} = 1, C_{ij} \geq 0.$$

- 3.4 Recalculate archetypes  $Y = X^T C$
  - 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 the Breiman's point on choosing initial mixtures (archetypes) we use the following method. The method proceeds by randomly selecting a data point as an archetype and selecting subsequent data points  $x_i$  (archetypes) the furthest away from already selected ones  $x_j$ . Such a new data point is selected according to

$$a^{new} = \arg \max_i \left\{ \sum_j \|x_i - x_j\|, j \in C \right\} \quad (3.3)$$

where  $\|\cdot\|$  is a given norm and  $C$  is a set of indices 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 [33].

*Example 1.* In order to show AA and another well know matrix factorization method in use we describe the following example. Assume  $X$  is an  $12 \times 12$  matrix. In NMF the non-negative matrix  $X$  is decomposed into two nonnegative matrices,  $N$  and  $M$ , as shown in Figure 3.1(a). In AA the matrix  $X$  is decomposed into two stochastic matrices,  $C$  and  $S$  as shown in Figure 3.1(b). In contrast to NMF, AA decomposes an input sparse matrix into two very sparse stochastic matrices. Figure 3.2 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. Figure 3.2 compares the sparseness of matrices obtained by AA and NMF. The non-negative matrices  $X$  in Figure 3.2 were a randomly generated  $n$ -by- $n$  matrices, and the values of  $n$  were set to 20, 50, 80, 100 and 200.

### 3.2.2 MDS problem statement and corpus modeling

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 the 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

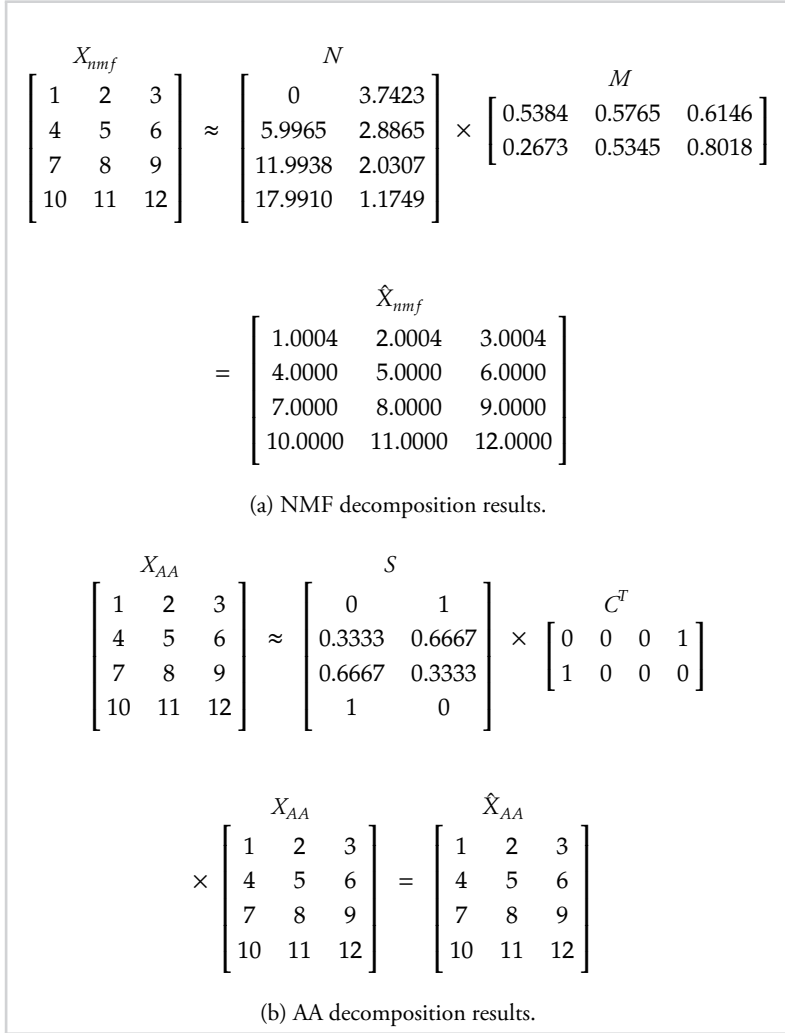


Figure 3.1  
Decomposition examples  
from NMF and AA.



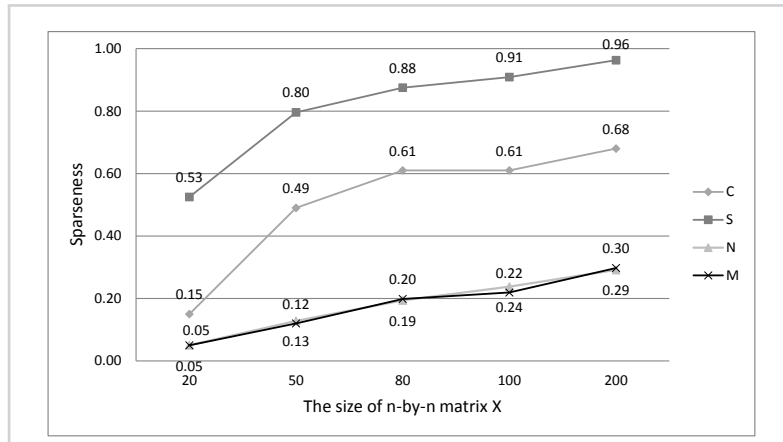


Figure 3.2

Comparison of sparseness between AA and NMF decomposition.

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 fulfills the above requirements. In our study, this goal has been reached by 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  $D = \{s_1, s_2, \dots, s_n\}$ , where  $n$  denotes the number of sentences,  $s_i$  denotes the  $i$ th 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  $T = \{t_1, t_2, \dots, t_m\}$  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:

$$tf_{iv} = freq_{iv} \quad (3.4)$$

where  $freq_{iv}$  is the frequency of term  $t_v$  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:

$$isf_{iv} = \log\left(\frac{n}{n_v}\right) \quad (3.5)$$

where  $n$  denotes the number of all sentences in the corpus, and  $n_v$  denotes the number of sentences that term  $t_v$  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-isf* weighting scheme can be formulated as:

$$w_{iv} = tf_{iv} \times isf_{iv} = freq_{iv} \times \log\left(\frac{n}{n_v}\right) \quad (3.6)$$

here the weight  $w_{iv}$  of a term  $t_v$  in a sentence  $s_i$  is defined by the product of the local weight of term  $t_v$  in sentence  $s_i$  and the global weight of term  $t_v$ . A 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,  $s_i = [w_{i1}, w_{i2}, \dots, w_{im}]$ , where  $w_{iv}$  is the weight of the term  $t_v$  in the sentence  $s_i$ . This measure is based on the angle  $\alpha$  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  $s_i = [w_{i1}, w_{i2}, \dots, w_{im}]$  and  $s_j = [w_{j1}, w_{j2}, \dots, w_{jm}]$  can be derived from the Euclidean dot product:

$$(s_i, s_j) = |s_i| \cdot |s_j| \cdot \cos \alpha \quad (3.7)$$

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  $\alpha$  between them. Given Eq. (3.7) the cosine similarity is therefore:

$$\text{sim}(s_i, s_j) = \cos \alpha = \frac{(s_i, s_j)}{|s_i| \cdot |s_j|} = \frac{\sum_{l=1}^m w_{il} w_{jl}}{\sqrt{\sum_{l=1}^m w_{il}^2 \cdot \sum_{l=1}^m w_{jl}^2}}, i, j = 1, 2, \dots, n. \quad (3.8)$$

*The sentence similarity matrix* describes a similarity between sentences presented as points in Euclidean space. Columns and rows are sentences while their intersection gives the similarity values of corresponding sentences calculated with Eq. (3.8).

*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. (3.6).

*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  $n$  and the number of terms  $m$ . Then  $T$  denotes the  $m \times n$  term-sentence matrix and a sentence to sentence similarity matrix may also be represented as a vector space, defining an  $n \times n$  matrix  $A$ . A 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,

$$[[TA]_{m \times n}^T]_{n \times m} = [[T]_{m \times n} \times [A]_{n \times n}]^T \quad (3.9)$$

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

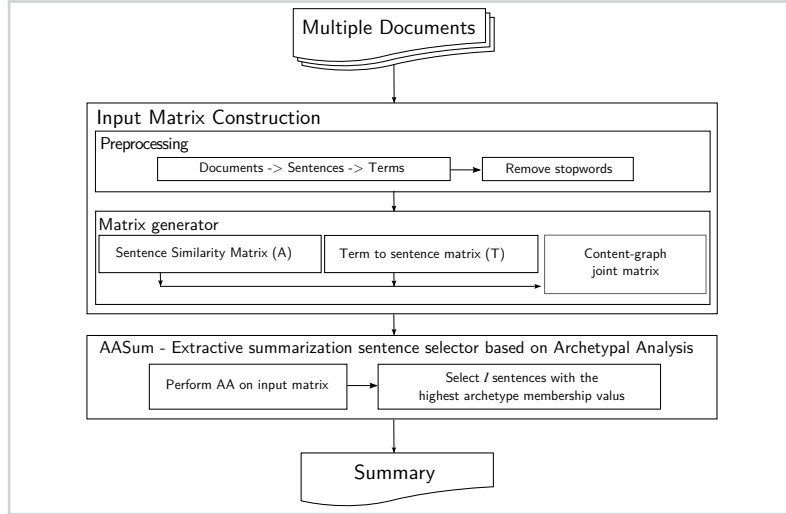


Figure 3.3

The overall illustration of the general multi-document summarization method based on archetypal analysis.

### 3.2.3 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, the 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 Figure 3.3. Here  $l$  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 documents from document set  $D$  into  $n$  individual sentences

without taking into consideration the ordering of documents.

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. (3.8).
  - ii Generate the term to sentence occurrence matrix  $T$  by using Eq. (3.6).
  - iii Return the matrix product of  $T$  and  $A$  using the Eq. (3.9).
4. Perform AA on matrix  $X$ .
  - i Estimate the decomposition matrices  $S, C$  and  $X^T C$  using the AA algorithm given in Section 3.2.1.
  - ii For each archetype  $i$  calculate its significance i.e. the sum of values in corresponding column of the matrix  $X^T C$ ,  $Sa_i = \sum_{j=1}^m X^T C_{ji}$ .
  - iii Sort the archetypes in decreasing order of significance, i.e. order the columns of matrix  $C$  based on values of  $Sa_i$ .
  - iv Eliminate  $\epsilon$  archetypes with lowest significance and return the result.
5. Select  $l$  sentences with the highest archetype membership values from the most significant archetypes.
  - i Start with the most significant archetype (the first column of the column-sorted matrix  $C$ ) and extract the sentence with the highest value in this column. Then continue with the second most significant archetype (the second column of  $C$ ) and so on. That is, sentences with the 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, i.e.  $sim(s_i, s_j) \geq 0.9$ , the newly selected sentence is not included in the summary.

Here,  $\epsilon$  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 columns of matrix  $C$ . 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.

#### 3.2.4 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 Figure 3.4. 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  $s_6$  of the second cluster. One can also argue that there is a topic drift in the first cluster occurring in the neighborhood of  $s_4$ .

Assume  $X$  is an  $12 \times 12$  matrix representing the similarity graph. By setting  $z = 3$

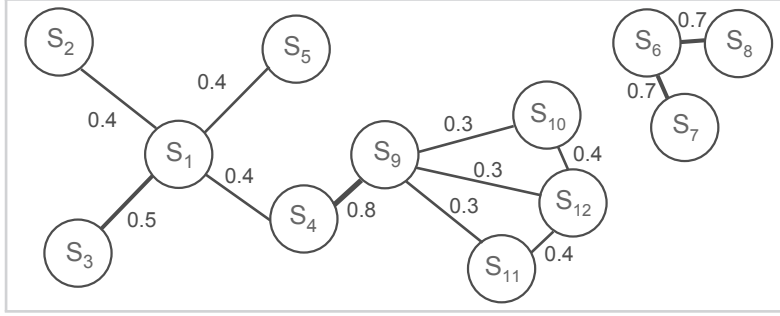


Figure 3.4

Sentence similarity graph, where nodes denote sentences and weighted edges represent the similarity between corresponding sentences.

we obtain matrices  $S^T$ ,  $C^T$  and  $X^T C$  estimated by AA as shown in Table 3.1. Decomposed matrices  $S^T$  and  $C^T$  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 column of the matrix  $X^T C$ , and then we order the archetypes based on the column sums. From the last column of Table 3.1, it can be seen that *Archetype*<sub>3</sub> is the most significant sentence archetype and it can be seen as “very-good” archetype while *Archetype*<sub>1</sub> is the least significant and it can be considered as “bad” from which no sentence should be extracted into the summary. From  $S_2^T$  in Table 3.1 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.  $C_3^T$ , 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  $C_2^T$ , 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  $C_1^T$  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.,  $S^T$  reflects the clustering into archetypes and  $C^T$  the rank

*Table 3.1*  
Results of Archetype Analysis on the illustrative example.

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>	S <sub>10</sub>	S <sub>11</sub>	S <sub>12</sub>	$\sum R_i$
$S_1^T$	0	0.7944	0.7944	I	0.7944	0	0	0	0	0.8785	0.8785	0.6834	
$S_2^T$	0.0711	0.1798	0.1798	0	0.1798	I	I	I	0	0.0235	0.0235	0.0240	
$S_3^T$	0.9288	0.0257	0.0257	0	0.0257	0	0	0	I	0.0979	0.0979	0.2925	
$C_1^T$	0	0.1239	0.1239	0.2493	0.1239	0	0	0	0	0.1460	0.1460	0.0877	
$C_2^T$	0	0.0155	0.0155	0	0.0155	0.3240	0.3100	0.3100	0	0	0	0	
$C_3^T$	0.4187	0	0	0	0	0	0	0	0.5313	0	0	0.0509	
$X^T C_1$	0.2734	0	0	0	0	0	0	0	0.3134	0.0438	0.0438	0.1460	0.8206
$X^T C_2$	0.0187	0	0	0	0	0.4453	0.2226	0.2226	0	0	0	0	0.9093
$X^T C_3$	0	0.1674	0.1674	0.6344	0.1674	0	0	0	0.0152	0.1848	0.1848	0.1593	1.6813



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  $\epsilon$  the least significant ones, the number of archetypes  $z$  may be set to be close to the number of sentences to be extracted plus the number  $\epsilon$ .

### 3.2.5 Discussions and Relations

Since various matrix decomposition methods such as PCA/SVD, k-means and NMF have been successfully employed in MDS, 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  $X \in \mathbb{R}_+^{n \times m}$  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  $L_1 \in \mathbb{R}_+^{n \times p}$ ,  $L_2 \in \mathbb{R}_+^{p \times m}$ ) i.e.

$$(L_1 L_2) = \arg \min_{L_1 L_2} \|P - L_1 L_2\|^2 \quad (3.10)$$

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

them in the decreasing order:

1. *AA*:

$$(CS) = \arg \min_{C,S} \|X - SC^T X\|^2$$

s.t.  $C$  is stochastic

$S$  is stochastic

2. *k-means*:

$$(CS) = \arg \min_{C,S} \|X - CS\|^2$$

s.t.  $C$  is stochastic

$S$  is binary

3. *NMF*:

$$(CS) = \arg \min_{C,S} \|X - CS\|^2$$

s.t.  $C$  is nonnegative

$S$  is nonnegative

4. *PCA/SVD*:

$$(CS) = \arg \min_{C,S} \|X - CS\|^2$$

s.t.  $C^T C = I$

where  $S$  and  $C$  are output or decomposition matrices,  $X$  is input or matrix to decompose, while stochastic, binary and nonnegative are constraints involved in each optimization problem. NMF, k-mean and SVD have been shown as successful decomposition methods 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 problems. Supporting evidences can be found in the next section where we compare AASum with other decomposition based summarization methods.

Table 3.2

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	12874	34539
Summary length	665 bytes	250 words

### 3.3 Experiments

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

#### 3.3.1 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<sup>1</sup> 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 3.2 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 publicly available stop list<sup>2</sup> and the terms were stemmed using the Porter's scheme<sup>3</sup>, which is a commonly used algorithm for word stemming in English.

The summarization evaluation methods can be divided into two categories: intrinsic and extrinsic [41, 42]. The intrinsic evaluation measures the quality of summaries directly (e.g., by comparing them to ideal summaries). The extrinsic methods measure

<sup>1</sup><http://duc.nist.gov>

<sup>2</sup><ftp://ftp.cs.cornell.edu/pub/smart/english.stop>

<sup>3</sup><http://www.tartarus.org/martin/PorterStemmer/>

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 the ROUGE metric. In our experiments, we used for evaluation the Recall Oriented Understudy for Gisting Evaluation (ROUGE) evaluation package [43], 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 the 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 the following formula [43]

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{Summ}_{ref}} \sum_{N\text{-gram} \in S} \text{Count}_{match}(N\text{-gram})}{\sum_{S \in \text{Summ}_{ref}} \sum_{N\text{-gram} \in S} \text{Count}(N\text{-gram})} \quad (3.11)$$

where  $N$  stands for the length of the N-gram,  $\text{Count}_{match}(N\text{-gram})$  is the maximum number of N-grams co-occurring in the candidate summary and the set of reference-summaries.  $\text{Count}(N\text{-gram})$  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-SU<sub>4</sub> [44]. ROUGE-L is the Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically. ROUGE-W is the Weighted LCS-based statistics that favors consecutive LCSes. ROUGE-S is the Skip-bigram based co-occurrence statistics. Skip-bigram is any pair of words in their sentence order. ROUGE-SU is the Skip-bigram plus unigram-based co-occurrence statistics.

### 3.3.2 Input matrix selection and its 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 the sentence similarity matrix, and are also known as graph based methods such as LexRank and DSQ. Note

Table 3.3

AASum methods comparison in input matrix construction phase on DUC2004 and DUC2006 for  $z = 2$ . Remark: \* indicates the best results in this set of experiments.

Summarizers	DUC2004		DUC2006	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
AASum-W <sub>1</sub>	0.3605	0.0683	0.3334	0.0441
AASum-W <sub>2</sub>	0.3706	0.0871	0.4239	0.0908
AASum-W <sub>3</sub>	0.4115*	0.0934*	0.4291*	0.0944*

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 the type of the input matrix we apply AA in three different ways during the summarization process. Each of the way, denoted as AASum-W<sub>1</sub>, AASum-W<sub>2</sub>, AASum-W<sub>3</sub> is discussed below. The AASum-W<sub>1</sub>, *the content based method*, performs on the term by sentence matrix formed by Eq. (3.6). The AASum-W<sub>2</sub>, *the graph based method*, performs on the sentence by sentence similarity matrix constructed by Eq. (3.8). The AASum-W<sub>3</sub>, *the content-graph method*, performs on the *joint matrix* of later two ones and it is obtained by using Eq. (3.9). Informally, we can look at the content-graph representation as saying that a sentence is connected to sentences that have  $[TA]^T$  term distribution. In order to better understand the results, we use Table 3.3 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 content-graph representation better describes the sentence relations. In the following we use only AASum-W<sub>3</sub> and refer to it merely as AASum.

### 3.3.3 Impact of the archetype algorithm's initialization on summarization performance and on the speed of the convergence

This section investigates whether (1) the summarization outcome and (2) the speed of the convergence, depends on the initialization of matrices  $C$  and  $S$ . As noted in Section 3.2.1, one can simply initialize the matrix  $C$  to data points selected at random without replacement from the input data. Initialization process then continues with computing  $S$  and  $X^T C$  given the  $C$ . Let us name this initialization method as *ran-*

Table 3.4

Impact of the initialization of the archetype algorithm on summarization for  $z = 2$ .

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

*dom*. Another initialization method presented in Section 3.2.1 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 3.4, suggest that the summarization outcome is not sensitive to the initialization method.

Similarly, to experimentally demonstrate the impact of initialization on the speed of convergence we designed the following experiment. We run AA algorithm sequentially 100 times with each of the initialization methods. Then, in a very straightforward way, the average execution time (in seconds) over all runs spent on calculation are computed and reported in Table 3.5. Each internal slot of the table reports the average execution time and the standard deviation of 100 runs in seconds. Results, presented in Table 3.5, suggest that the speed of convergence is significantly dependent on initialization method. It can be observed that this dependency is in positive correlation with the number of archetypes. The higher mean and standard deviation results of random initialization are as it is expected.

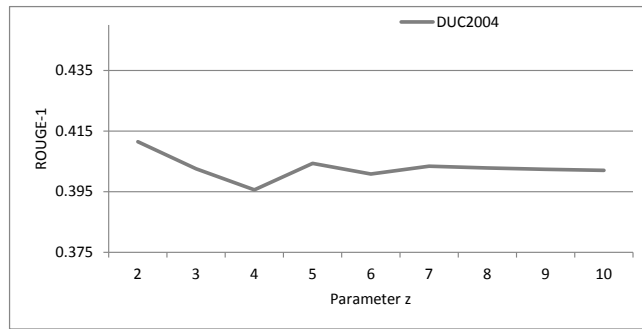
### 3.3.4 Impact of the number of archetypes

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

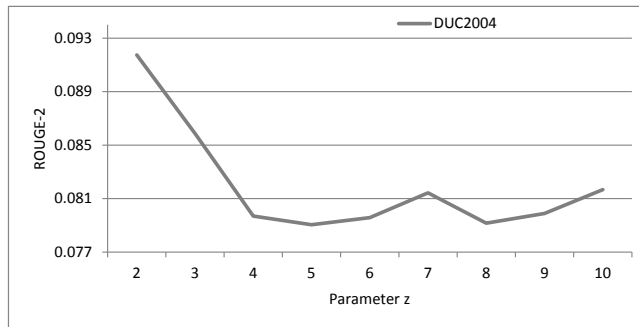
Table 3.5

Impact of the initialization of the archetype algorithm on the speed of convergence.

Initialization	# of archetypes			
	2	8	16	32
<i>random</i>	5.8/4.8	21.0/7.9	35.9/14.1	55.8/8.1
<i>f-away</i>	8.0/1.0	18.6/2.4	23.4/7.3	29.2/5.0



(a) ROUGE-1 vs  $z$  on DUC2004



(b) ROUGE-2 vs  $z$  on DUC2004

Figure 3.5

Impact of archetype number on summarization.

Table 3.6

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

number of archetypes  $z$ . A simple approach for choosing the value of  $z$  is to run the algorithm for different numbers of  $z$  where the selection criteria should be the maximization of the summary evaluation outcomes. In previous experiments, the archetype number  $z$  is set to be close to the number of sentences to be extracted plus the number  $\epsilon$ . The  $\epsilon$  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  $z$ . We gradually increase the value of  $z$ , in the range from 2 to 10 and the results show that increasing the number of extracted archetypes do not necessary increases the summarization performance. Figure 3.5 plots the ROUGE-1 and ROUGE-2 curves of our AA based approach on the DUC2004 dataset.

### 3.3.5 Comparison with related methods

We first report the standalone performance results of the proposed method in Table 3.7 where the mean value as well as 95% confidence interval over all topics of the recall,



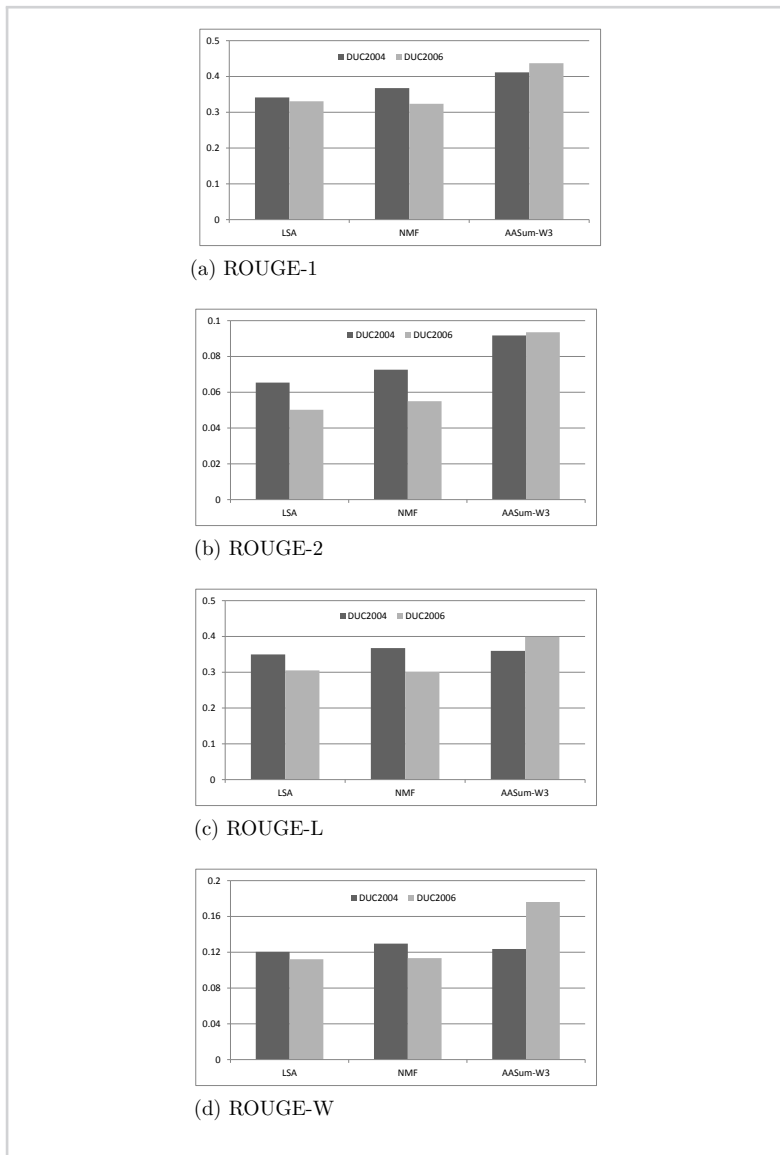


Figure 3.6

Overall summarization performance on DUC2004 and DUC2006 data for  $z = 2$ .

Table 3.7

General evaluation of the AASum on the DUC2004 and DUC2006 datasets for  $z = 2$ .

	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]

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 [15], LSA and NMF are two competing matrix decomposition techniques for the task of MDS. From Figure 3.6 we can see that NMF shows better performance than LSA. This is in consistency with results reported in [15] and it can be mainly contributed to the property of NMF to select more meaningful sentences by using the more intuitively interpretable semantic features and by better grasping the innate structure of documents. Our proposed approach shows even better performance, see Figure 3.6. This is because it uses the archetypal analysis to detect the archetypal structure which can cluster and rank sentences more effectively than above-mentioned

Table 3.8

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	0.3308	0.0502	0.1023
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-W <sub>3</sub>	0.4115*	0.0934	0.1376*

approaches. Figure 3.6 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 3.6. 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 3.8 and Table 3.9. It produces very competitive results, which apparently outperforms many of the methods in both years. More importantly, 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 [17]. 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 3.9

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-W <sub>3</sub>	0.4291*	0.0944	0.1680

### 3.4 Conclusion and future work

The main contributions of the chapter are the following:

- (i) The chapter 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 the 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 chapter also shows how AA can be used for simultaneously sentence clustering and ranking.
- (iv) This chapter 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 chapter 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 summarization 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.





*Weighted Archetypal Analysis  
of the multi-element graph for  
query-focused multi-document  
summarization*

The content of the chapter is partially based on our work [45], where the query-oriented multi-document summarization problem is treated by using the weighted archetypal analysis of the multi-element graph model.

### 4.1 Introduction

The Query-focused multi-document summarization is a special case of multi-document summarization. Given a query, the task is to produce a summary which can respond to the information required by the query. Different from generic summarization, which needs to preserve the typical semantic essence of the original document(s) [41, 46], query-focused summarization purposely demands the most typical (archetypal) summary biased toward an explicit query.

The continuing growth of available online text documents makes research and applications of query-focused document summarization very important and consequently attracts many researchers. Since it can produce brief information corresponding to the users queries, it can be applied to various tasks for satisfying different user interests. The queries are mostly real-world complex questions (e.g., "Track the spread of the West Nile virus through the United States." is a query example). Such complicated questions make the query focused summarization task quite difficult. The real problem is how to model the question jointly with the documents to be summarized and thus bias the answer, i.e. summary, towards the provided question.

Most existing research on applying matrix factorization approaches to Q-MDS explores either low rank approximation or soft/hard clustering methods. The former have a great degree of flexibility but the features can be harder to interpret. The latter extract features that are similar to actual data, making the results easier to interpret, but the binary assignments reduce flexibility. These techniques can be jointly seen as a factor analysis description of input data exposed to different constraints. Inadequately, most of these methods does not directly incorporate the query information into summarization process, thus the summarization is general about the document collection itself. Moreover, most existing works assume that documents related to the query only talk about one topic. Even though query-focused summarization, by its definition, is biased toward a given query, in our understanding it doesn't mean that the produced summary should not show the diversity in content as much as possible.

In this chapter, we try to overcome limitations of the existing algebraic methods and study a new setup of the problem of query-focused summarization. Since the



archetypal analysis completely assembles the advantages of clustering and the flexibility of matrix factorization we propose using the AA in Q-MDS. Consequently, the main concerns of this chapter are: (1) how to incorporate query information in its own nature of an archetypal analysis based summarizer; and (2) how to increase the variability and diversity of the produced query-focused summary. For the first concern, we propose a weighted version of archetypal analysis based summarizer able to directly use the query information. The second one is answered by the nature of the archetypal analysis itself, which clusters the sentences into distinct archetypes.

The main contributions of the chapter are three-fold:

1. A novel query-focused summarization method wAASum is proposed.
2. Modeling the input documents and query information as a multi-element graph is introduced.
3. The effectiveness of the proposed approach is examined in the context of Q-MDS.

To show the efficiency of the proposed approach, we compare it to other closely related summarization methods. We have used the DUC2005 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 chapter is organized as follows: Section 4.2 introduces the weighted archetypal analysis, whereas Section 4.3 presents the multi-element graph modeling. The details of the proposed summarization approach wAASum are presented in Section 4.4, where we give an overview of the new approach and an illustrative example of its use. Section 4.5 shows the evaluation and experimental results. Finally, we conclude in Section 4.6.

## 4.2 *Weighted Archetypal Analysis*

In the first archetypal problem defined by Eq. (3.2), each data point and hence each residual participates to the minimization with the same weight. Note that  $X$  is an  $n \times m$  input matrix and let  $W$  be a complementing  $n \times n$  square weight matrix. The weighted version of the archetypal problem can then be formulated as the minimization of

$$RSS(k) = \|W(X - SY^T)\|^2 \text{ with } Y = X^T C \quad (4.1)$$

Since weighting the residuals is identical to weighting the data set:

$$\begin{aligned} W(X - SY^T) &= W(X - S(X^T C)^T) \\ &= W(X - SC^T X) \\ &= WX - W(SC^T X) \\ &= WX - (WS)(C^T W^{-1})(WX) \\ &= \hat{X} - \hat{S}\hat{C}\hat{X} = \hat{X} - \hat{S}\hat{Y}^T \end{aligned}$$

the problem can be rewritten as minimizing

$$\begin{aligned} RSS(k) &= \|\hat{X} - \hat{S}\hat{Y}\|^2 \\ \text{with } \hat{Y} &= \hat{C}\hat{X} \text{ and } \hat{X} = WX \end{aligned} \quad (4.2)$$

This reformulation provides a way to use the original algorithm with the supplementary pre-processing step to calculate  $\hat{X}$  and the additional post-processing step to recalculate  $S$  and  $C$  for the data set  $X$  given the archetypes  $\hat{Y} = C\hat{X}$ . Calculating the  $\hat{X}$  requires standardizing the input matrix  $X$  before and again converting it to its original form after the calculation. Also, recalculating the  $S$  and  $C$  requires solving several convex least squares problems. The weight matrix  $W$  can formulate different intentions. In our study  $W$  is a diagonal matrix of the weights representing the sentences to query similarity.

### 4.3 Multi-element Graph Model

In this section we present our approach to modeling the Q-MDS problem. The graph model presented in this study is motivated by the observed evidences that (1) building a joint model of sentence contents and connections produces a better model than that built from contents or connections alone [47]; (2) introducing the notion of document into graph model and distinguishing intra-document and inter-document sentence relations can visibly improve the summarization results [20]. Based on these studies, we believe the graph model for Q-MDS should describe the following elements and their relations: terms, sentences, documents and the given query.

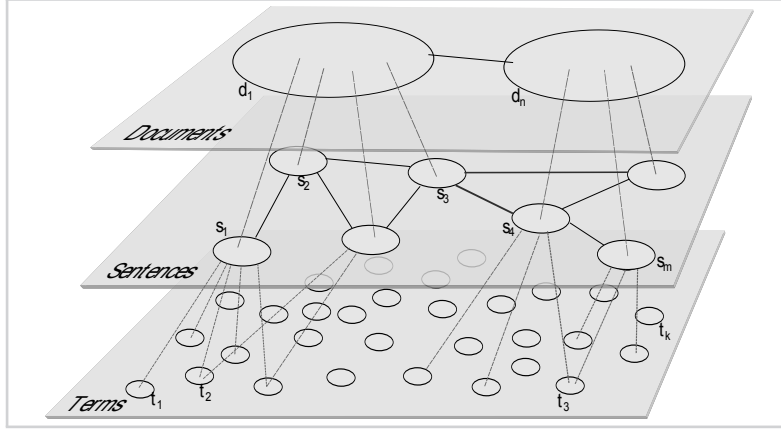


Figure 4.1

Multi-element similarity graph, where the first level represents the document, the second level denotes the sentence and the third level represents the term relation graph.

Let a set of documents  $D$  be represented as a text similarity graph  $G = (V_t, V_s, V_d, E^{V_t}, E^{V_s}, E^{V_d}, \alpha_v, \beta_v, \gamma_v, \alpha_e, \beta_e, \gamma_e)$ , where  $V_t, V_s$  and  $V_d$  represent the term, sentence and document vertex sets, respectively.  $E^{V_t} \subseteq V_t \times V_s, E^{V_s} \subseteq V_s \times V_s$  and  $E^{V_d} \subseteq V_d \times V_d$  are term-sentence, sentence and document edge sets.  $\alpha_v : V_t \rightarrow \mathfrak{R}_+, \beta_v : V_s \rightarrow \mathfrak{R}_+$  and  $\gamma_v : V_d \rightarrow \mathfrak{R}_+$  are three functions defined to represent term, sentence and document vertices, while  $\alpha_e : E^{V_t} \rightarrow \mathfrak{R}_+, \beta_e : E^{V_s} \rightarrow \mathfrak{R}_+$  and  $\gamma_e : E^{V_d} \rightarrow \mathfrak{R}_+$  are functions for assigning term, sentence and document edges. Figure 4.1 illustrates the proposed graph model where in addition to conventional similarity graph, one can observe the following important information: (1) a methodical way of combining information from the terms and documents on one side and sentence similarity connection structure on the other side; (2) the containment relation between terms and the sentence they belong to; (3) the containment relation between sentences and the document they originate from; (4) the similarity relation among documents; (5) the sentence to sentence similarity relations divided into two categories, i.e. the one within the document and the one cross over two documents. The containment relation between sentences and documents and the discrimination of the sentence to sentence similarities to two different types is realized through weighting schema presented in the following paragraphs. As for Q-MDS, an additional kind of object (i.e. query) is involved.

Recall that our final purpose is sentence ranking for selecting the summary sentences.

Table 4.1

Labeling functions in graph and matrix notation;  $n$ ,  $m$  and  $k$  are respectively the total number of the documents, sentences and terms in a document set.

Labeling function		
Graph Notation	Matrix Notation	Graph/matrix type
$\alpha(s_i, s_j) = \frac{sim(s_i, s_j)}{\sum_{s_k \in D \cap k \neq i} sim(s_i, s_k)}$	$A = [\alpha(s_i, s_j)]_{m \times n}$	Sentence similarity
$\beta(d_i, d_j) = \frac{sim(d_i, d_j)}{\sum_{d_k \in D \cap k \neq i} sim(d_i, d_k)}$	$B = [\beta(d_i, d_j)]_{y \times y}$	Document similarity
$\gamma(s_i, t_j) = tf(s_i, t_j) \times isf(s_i, t_j)$	$G = [\gamma(s_i, t_j)]_{m \times n}$	Term to sentence
$\delta(s_i, q) = \frac{sim(s_i, q)}{\sum_{s_k \in D} sim(s_k, q)}$	$W = [\delta(s_i, q)]_{m \times m}$	Sentence to query diagonal matrix

Therefore the term and the document-level information in the proposed model are not used directly to evaluate the sentence-level nodes and edges, but to regulate their weights and to enrich the sentence similarity graph model with sentence contents.

To reflect the impact of the document dimension on the sentence similarity matrix, the sentence edges that connect different documents are additionally weighted by the similarity between the documents they connect. Hence in order to show bias toward the cross-document sentence edges, the weight matrix  $W_M$  is introduced. Typically, the elements in  $W_M$  which connect sentences from the same document are set to 1. In this way they denote the relative weight of the intra-document sentence edges. On the contrary, elements which connect sentences from different documents are defined by the relations between the two corresponding documents. In this context it is defined as  $W_M(i, j) = 1 + \beta(d(s_i), d(s_j))$  where  $d(s)$  represents the document that contains the sentence  $s$ .

Moreover in the interest of reflecting the impact of the term dimension on the sentence similarity matrix, i.e. to obtain the content-graph representation, we compute

the inner product of the sentence to sentence and the term to sentence matrices.

In the presented model the sentence edge function is formulated as the normalized similarity between the two sentences  $s_i$  and  $s_j$ , and the document edge function is defined as the normalized similarity between the two documents  $d_i$  and  $d_j$ . Alike, the vertex function is biased to the relevance of the sentences to the query. The term to sentence edge function is formulated as the frequency of terms that occur in sentences times the inverse term frequency. For the compact list of the labeling functions in a graph and matrix notation see Table 4.1. The similarity between any two sentences (or documents) is defined as the cosine similarity between them, which is the most popular one used in information retrieval and text mining. The matrix  $W$  from Table 4.1 is later used as the weight matrix in weighted archetypal analysis summarization method (wAASum).

So, to summarize, the proposed graph model in matrix notation is:

$$[X]_{n \times m}^T = \left[ [W_M]_{m \times m} \odot [A]_{m \times m} \right]_{m \times m} \otimes [G]_{m \times n} \quad (4.3)$$

where,  $\odot$  denotes the Hadamard product and  $\otimes$  denotes the inner product.

#### 4.4 Query-focused document summarization by wAASum

In this section, a method for generating the Q-MDS summary by selecting sentences using the wAA is presented.

##### 4.4.1 wAASum

We give an overall illustration of the method in Figure 4.2. The main idea of the method is simple: sentences are soft-clustered into weighted 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 method, wAASum, consists of the following steps:

1. Construct the input matrix  $X$  using the Eq. (4.3).
2. Generate the input diagonal weight matrix  $W$  by using the labeling function  $\delta(s_i, q)$  given in the last row of the Table 4.1.
3. Perform weighted AA on matrix  $X$  given the  $W$ .

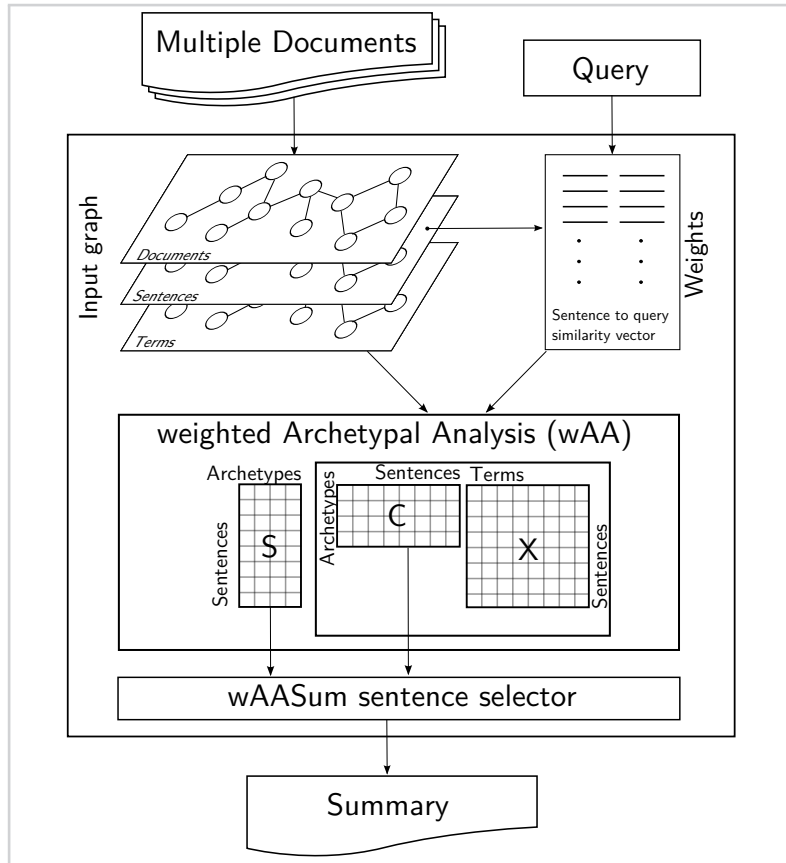


Figure 4.2  
The overall illustration of the query-focused multi-document summarization method based on weighted archetypal analysis.

- i Estimate the factorization matrices  $C$  and  $S$  as described in Section 3.2.1.
  - ii For each archetype  $i$  calculate its significance  $Sa_i$ , i.e. the sum of values in the corresponding column of the matrix  $X^T C$ ,  $Sa_i = \sum_{j=1}^m X^T C_{ij}$ .
  - iii Sort the archetypes in the decreasing order of significance, i.e. order the columns of matrix  $C$  based on values of  $Sa_i$ .
  - iv Eliminate  $\epsilon$  archetypes with lowest significance and return the result.
4. Select  $l$  sentences with the highest weighted archetype membership value.
    - i Start with the most significant archetype and extract sentences in the order according to their values in  $C$ . That is, sentences with the highest archetype membership value in each archetype (column of matrix  $C$ ) are selected and if the summary length is not met then the extraction step continues with the second highest values, and so on.
    - ii Each selected sentence is compared to previously selected ones and if there is a significant similarity between them, i.e.  $sim(s_i, s_j) \geq 0.9$ , the newly selected sentence is not included in the summary.

In the above algorithm, the third and fourth steps are crucial. Our purpose is to cluster sentences into weighted archetypes and subsequently select the sentences with the highest archetype membership weights. In the third step, the significant sentences are more likely to be clustered into weighted archetypes with high significance. Since the sentences with the higher membership values are ranked higher, the sentences selected by the fourth step are the most central ones. The fourth 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 aspects of the summarization, namely the relevance and the content coverage. The last important effect 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 (4.ii) of above algorithm. In the following subsection we present the usage of wAASum on an illustrative example.

Table 4.2

Results of Weighted Archetype Analysis on the illustrative example from Figure 4.3.

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$	$\sum R_i$
$C_1^T$	0.2462	0	0	0	0	0	0	0	0.7336	0	0	0	
$C_2^T$	0.0108	0.0043	0.0043	0	0.0043	0.9385	0	0	0	0.0189	0.0189	0	
$C_3^T$	0.0035	0.0577	0.0577	0.8645	0.0577	0	0	0	0	0	0	0.0038	
$C_4^T$	0	0	0	0	0	0	0	0	0	0.0377	0.0377	0.9246	
$C_5^T$	0.0613	0.1443	0.1443	0	0.1443	0	0.2259	0.2259	0	0.0270	0.0270	0	
$S_1^T$	0.2646	0	0	0.0122	0	0.0026	0.0577	0.0577	0.5833	0.0077	0.0077	0.0064	
$S_2^T$	0	0.0045	0.0045	0.0154	0.0045	0.6558	0.1081	0.1081	0.0895	0	0	0.0097	
$S_3^T$	0	0.1002	0.1002	0.5004	0.1002	0	0	0	0.0057	0.0967	0.0967	0	
$S_4^T$	0	0	0	0.0283	0	0.0064	0.1166	0.1166	0.1547	0.0264	0.0264	0.5245	
$S_5^T$	0	0.0829	0.0829	0	0.0829	0.0043	0.3387	0.3387	0.0639	0	0	0.0056	
$X^T C_1$	0.0321	0.0898	0.0898	0.4573	0.0898	0.0321	0.0321	0.0321	0.0326	0.1645	0.1645	0.1653	1.3819
$X^T C_2$	0.0161	0.0153	0.0153	0.0162	0.0153	0.0116	0.5692	0.5692	0.0218	0.0116	0.0116	0.0253	1.2985
$X^T C_3$	0.4350	0	0	0	0	0	0	0	0.5958	0	0	0.0117	1.1181
$X^T C_4$	0	0	0	0	0	0	0	0	0.2849	0.3493	0.3493	0.0341	1.0598
$X^T C_5$	0.1412	0.0422	0.0422	0.0463	0.0422	0.2363	0.0258	0.0258	0.0373	0.0258	0.0258	0.0411	0.7319



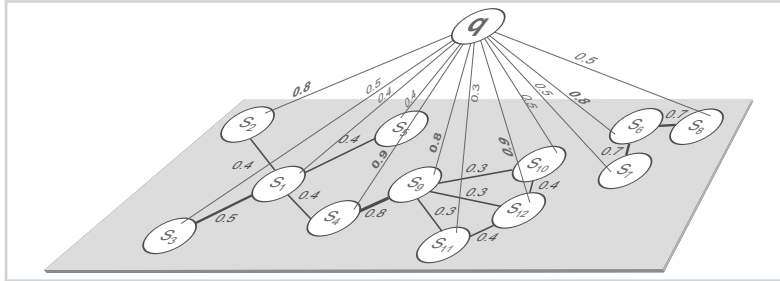


Figure 4.3

Sentence similarity graph, where nodes denote sentences and weighted edges represent the similarity between corresponding sentences. Additionally each sentence is also weighed with its similarity to the given query.

#### 4.4.2 An illustrative example

In order to demonstrate the advantages of wAA as the method for simultaneous sentence clustering and ranking with the respect to the sentence weights, a simple example is given in Figure 4.3. We present the synthetic data set as an undirected sentence similarity graph, where nodes denote sentences and edges represent similarity between connected nodes. Additionally, to visually give the gist of the sentence node weights in Figure 4.3 each sentence is also connected to the given query with the edge weighted by the corresponding sentence to query similarity. Looking at the graph, one can observe two clusters of sentences, where  $\{s_1, s_9\}$  are the central sentences of the first and  $s_6$  of the second cluster. Nevertheless, the observer should also pay attention to the nodes similarity to the query  $q$ . With this lead, even though the node  $s_1$  is a highly connected sentence, the potential of it to be selected as significant sentence largely decreases when the dimension of query-similarity is introduced.  $s_1$ 's similarity to given query  $q$  is quite low (0.4). On the other hand the potential of  $s_4$  increases by the same reasoning. Even though  $s_4$  does not show significant centrality in the original graph, its similarity to the given query  $q$  is very high (0.9).

Assume  $X$  is an  $12 \times 12$  matrix representing the similarity graph. By setting  $z = 5$  we obtain matrices  $C^T$ ,  $S^T$  and  $X^T C$  estimated by wAA as shown in Table 4.2. Factorized matrices  $S^T$  and  $C^T$  can be interpreted as clustering and ranking outputs, respectively. The extracted archetypes, five of them, are in fact weighted 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

Table 4.3

Experimental data description.

	DUC 2005	DUC 2006
Number of clusters	50	50
Avarage number of documents per set	31.86	25
Summary length	250 words	250 words

according to their significance we first calculate the sum of the each row of the matrix  $X^T C$ , and then we order the archetypes based on the row sums. From the last column of Table 4.2, it can be seen that  $Archetype_1$  is the most significant sentence archetype and it can be seen as the "best" archetype while  $Archetype_5$  is the least significant and it can be considered as the "worst", from which no sentence should be extracted into the summary. As expected, from  $C_1^T, C_2^T, C_3^T$  and  $C_4^T$  in Table 4.2 it can be seen that  $\{s_9, s_6, s_4, s_{12}, s_1\}$  belong to the first four clusters, the "good" archetypes, while the rest of the sentences belong to the other archetype with various values of memberships.  $C_1^T$ , the "best" archetype, shows that  $s_9$  has the highest ranking value, therefore it should be extracted into the resulting summary. From  $C_2^T$ , the second "best" archetype,  $s_6$  should be extracted, and from  $C_3^T$ , the third "best" archetype,  $s_4$  should be extracted to the resulting summary. This example shows that output matrices produced by wAA describe the data structure well and in various ways, i.e., matrix  $S^T$  reflects the clustering into archetypes and  $C^T$  the rank within each archetype. It is a non-trivial problem to choose the best number of archetypes to be estimated by wAA. But since we only select one representative sentence from each archetype starting from the most significant and not including  $\epsilon$  the least significant ones, the number of archetypes  $z$  may be set close to the number of sentences to be extracted plus the number  $\epsilon$ .

### 4.5 Experiments

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

Table 4.4

Summarization systems.

System ID	Description
wAASum	Our method
LSA	Latent semantic analysis, Matrix factorization
NNF	Non-negative matrix factorization
SNMF	Clustering, Matrix factorization
Biased-Lexrank	Graph-based
DrS-Q	Document sensitive graph-based
DDS	Constraint driven optimization
MCLR	Multi-objective optimization
Avg-Human	Average human performance from DUC
Avg-System	Average systems performance from DUC

#### 4.5.1 Experimental data and evaluation metric

We use the DUC2005 and DUC2006 data sets to evaluate our proposed method empirically, where benchmark data sets are from DUC<sup>1</sup> for automatic summarization evaluation. DUC2005 and DUC2006 data sets consist of 50 topics. Table 4.3 gives a brief description of the data sets. The task is to create a summary of no more than 250 words. In those document data sets, stop words were removed using the stop list<sup>2</sup> and the terms were stemmed using the Porter's scheme<sup>3</sup>, which is a commonly used algorithm for word stemming in English.

As in previous chapter, we used for evaluation the Recall Oriented Understudy for Gisting Evaluation (ROUGE) evaluation package [43] (see Eq. (3.11)). Here, we report the mean value over all topics of the recall scores of ROUGE-1, ROUGE-2 and

<sup>1</sup><http://duc.nist.gov>

<sup>2</sup><ftp://ftp.cs.cornell.edu/pub/smart/english.stop>

<sup>3</sup><http://www.tartarus.org/martin/PorterStemmer/>

Table 4.5

wAASum methods comparison for the input graph modeling phase on DUC2005 and DUC2006. Remark: \* indicates the best results in the corresponding set of experiments.

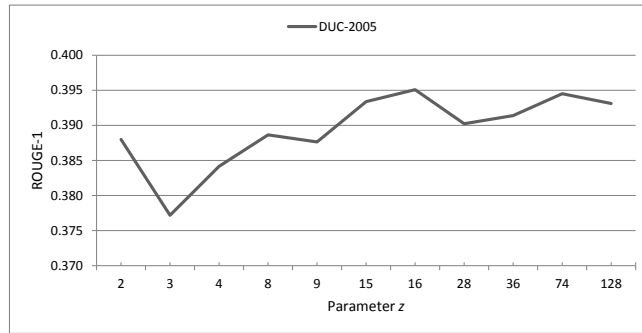
Summarizers	DUC2005			DUC2006		
	ROUGE-1	ROUGE-2	ROUGE-SU	ROUGE-1	ROUGE-2	ROUGE-SU
wAASum-W1	0.3470	0.0580	0.1157	0.3722	0.0632	0.1343
wAASum-W2	0.3790	0.0735	0.1365	0.4075	0.0872	0.1631
wAASum-W3	0.3896	0.0762	0.1392	0.4188	0.0887	0.1651
wAASum-W4	0.3945*	0.0797*	0.1420*	0.4238*	0.0917*	0.1671*

ROUGE-SU (skip bigram) [44].

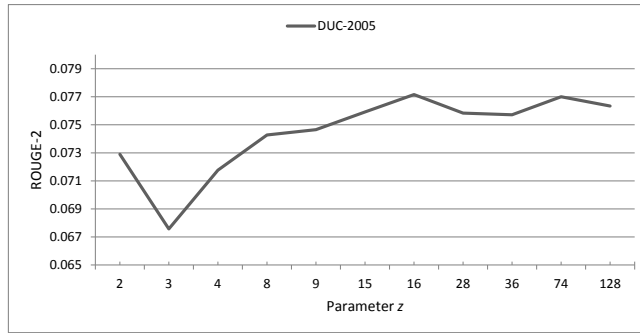
#### 4.5.2 Multi-element graph modeling and its impact on summarization

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.

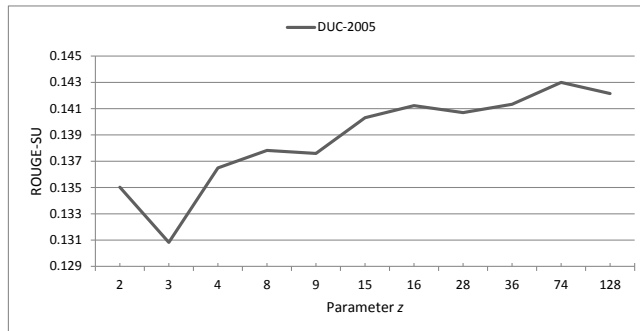
In the experiments, we compare the wAASum method's summarization results with respect to the input matrix type. Depending on a type of the input matrix we apply wAASum in four different ways during the summarization process denoted as wAASum-W1 to wAASum-W4. The wAASum-W1, the *content based method*, performs on the term by sentence matrix. The wAASum-W2, the *graph based method*, performs on the sentence by sentence similarity matrix. The wAASum-W3, the *content-graph method*, performs on the joint matrix of both previously mentioned matrices. Informally, we can look at the content-graph representation as saying that a sentence is connected to sentences that have  $[T \times A]^T$  term distribution. The wAASum-W4 as described in Sec. 4.3, is our novel *multi-element graph modeling* approach to the Q-MDS. In order to better understand the results, we use Table 4.5 to illustrate the comparison. The results clearly show that wAASum-W4 method performs best. This is due to facts comprehensively presented in Section 4.3. In the following only wAASum-W4 is used and denoted simply as wAASum.



(a) ROUGE-1 vs  $z$  on DUC2005



(b) ROUGE-2 vs  $z$  on DUC2005



(c) ROUGE-SU vs  $z$  on DUC2005

Figure 4.4

Impact of the archetype number  $z$  on query focused multi-document summarization.

### 4.5.3 Comparison with related methods

We compare wAASum with the most relevant methods to examine the effectiveness of the method for summarization performance improvement. For the detailed list of methods, see Table 4.4. These summarization methods are selected as the most widely used algebraic summarization methods. Although there are, for each year, more than 30 systems that have participated in the DUC competition, here we only compare with the DUC human average and the DUC system average result.

The input parameter of the wAASum algorithm, namely the number of archetypes is set to  $z = 16$ .

Tables 4.6 and 4.7 show the ROUGE scores of different methods using DUC2005 and DUC2006 data sets, respectively. The higher ROUGE score indicates the better summarization performance. The number in parentheses in each table slot shows the ranking of each method on a specific data set.

As indicated in [15], LSA and NMF are two competing matrix factorization techniques for the task of Q-MDS. From Tables 4.6 and 4.7 we can see that NMF shows better performance than LSA. This is in consistency with results reported in [15] and it can be mainly contributed to the property of NMF to select more meaningful sentences by using more intuitively interpretable semantic features and by better grasping the innate structure of documents. Our proposed approach shows even better performance. This is due to the weighted archetypal analysis, which can detect the archetypal structure with respect to a given query, and hence cluster and rank sentences more effectively than above mentioned approaches. Since LSA, NMF and SNMF are matrix factorization methods, the improvement of wAASum compared with them can be mainly attributed to wAA's ability to combine the clustering and the matrix factorization. The advantages of our approach are clearly demonstrated in Tables 4.6 and 4.7. It produces very competitive results, which apparently outperforms many of the methods in both years. More important, it is the best automatic system in DUC2006, and the second best in DUC2005. Notice also that all the results of wAASum are produced based on a simple similarity measure.

### 4.5.4 Impact of the number of archetypes

This problem is the same as the problem of choosing the number of components in other matrix factorization approaches and there is no rule for defining the correct num-

Table 4.6

Evaluation of the methods on the DUC2005 dataset. Remark: “-” indicates that the method does not officially report the results.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU <sub>4</sub>
Avg-Human	0.4417 (1)	0.1023 (1)	0.1622 (1)
Avg-DUCo5	0.3434 (7)	0.0602 (7)	0.1148 (6)
NMF	0.3110 (8)	0.0493 (8)	0.1009 (8)
LSA	0.3046 (9)	0.0407 (9)	0.1032 (7)
SNMF	0.3501 (6)	0.0604 (6)	0.1229 (5)
Biased-Lex	0.3861 (4)	0.0753 (5)	0.1363 (4)
DrS-Q	0.3785 (5)	0.0771 (4)	0.1337 (5)
DDS	0.3956 (2)	0.0852 (2)	0.1434 (2)
MCLR	-	-	-
wAASum	0.3945 (3)	0.0797 (3)	0.1420 (3)

ber of archetypes  $z$ . A simple approach for choosing the value of  $z$  is to run the algorithm for different numbers of  $z$  where the selection criteria should be the maximization of the summary evaluation outcomes. In experiments from Section 4.5.2 (and results in Table 4.5), the archetype number  $z$  was set to be close to the number of sentences to be extracted plus the number  $\epsilon$ .  $\epsilon$  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  $z$ . We gradually increase the value of  $z$ , in the range from 2 to 128 and the results show that increasing the number of extracted archetypes does not necessarily increase the summarization performance. The best results are observed for  $z = 16$ . Figure 4.4 plots the ROUGE-1, ROUGE-2 and ROUGE-SU curves of wAASum on the DUC2005 dataset.

#### 4.6 Conclusion and future work

This chapter has formalized the problem of query-focused document summarization as the weighted archetypal analysis problem. Additionally, it has presented our study of how to incorporate query information in the own nature of AA and how to use the weighted version of AA for simultaneous sentence clustering and ranking. We have ex-

Table 4.7

Evaluation of the methods on the DUC2006 dataset.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU <sub>4</sub>
Avg-Human	0.4576 (1)	0.1149 (1)	0.1706 (1)
Avg-DUCo6	0.3795 (7)	0.0754 (7)	0.1321 (7)
NMF	0.3237 (9)	0.0549 (8)	0.1061 (8)
LSA	0.3307 (8)	0.0502 (9)	0.1022 (9)
SNMF	0.3955 (5)	0.0854 (5)	0.1398 (4)
Biased-Lex	0.3899 (6)	0.0856 (4)	0.1394 (5)
DrS-Q	0.3955 (4)	0.0899 (3)	0.1427 (3)
MCLR	0.3975 (3)	0.0850 (6)	0.1385 (6)
DDS	-	-	-
wAASum	0.4238 (2)	0.0917 (2)	0.1671 (2)

amined the proposed method on several input matrix modeling configurations, where the chapter reports the best results on the multi-element graph model. The work presented in this chapter has proven that wAASum is an effective summarization method. Experimental results on the DUC2005 and DUC2006 datasets demonstrate the effectiveness of the proposed approach, which compares well to most of the existing matrix factorization methods in the literature. We think that wAASum has the potential to achieve further improvements in its performance on the query-focused summarization by incorporating the use of the query information in a more effective way.

We believe that in the future the performance of wAASum would possibly be further improved. There are many potential directions for improvements of wAASum, such as employing sophisticated methods for the query processing/expansion techniques or using the semantic similarity measure. Our future work will also apply the presented method to other summarization tasks.



*Weighted Hierarchical  
Archetypal Analysis based  
generic multi-document  
summarization framework*

5

## 5.1 Introduction

Different from generic summarization, which needs to preserve the typical semantic essence of the original document(s) [41, 48], the query-focused summarization purposely demands the most typical (archetypal) summary biased toward an explicit query. Lately, new summarization tasks such as the comparative summarization [3], and the update summarization [49] have also been proposed. The comparative summarization targets to summarize the dissimilarities between corresponding document groups, and the update summarization focuses on producing very brief summaries of the latest documents to apprehend novel information distinct from earlier documents.

In this chapter, we propose a new framework for MDS using the weighted hierarchical Archetypal Analysis (wHAASum). Many known summarization tasks, including generic, query-focused, update, and comparative summarization, can be modeled as different versions acquired from the proposed framework. An effective foundation to settle affinities among different summarization tasks while promoting their differences are served by this framework.

In our summarization framework, the generic MDS problem is firstly generalized to the weighted Hierarchical Archetypal Analysis problem. Then several useful properties of the wHAA are identified and taken into consideration for the greedy summarization algorithm. The latter is further shown to have the ability of addressing the MDS problem. We finally use this algorithm to propose the framework for different MDS tasks.

Our work described in this chapter, displays benefits from two perspectives:

1. it proposes a new generic framework to address different summarization problems;
2. it proposes a novel version of the well-known archetypal analysis algorithm, namely the weighted hierarchical archetypal analysis algorithm.

To the best of our knowledge, the problem of hierarchical wAA has not been proposed or studied before. Therefore, another significant contribution of this work is the presentation of the wHAA technique with its application to summarization.

The rest of the chapter is organized as follows. After introducing the original archetypal analysis(AA) and weighted archetypal analysis (wAA) algorithms and after proposing the novel hierarchical version of wAA in Section 5.2, we describe the hierarchical

wAA based summarization method in Section 5.3. Section 5.4 presents the framework for MDS, and shows how to model the four aforementioned summarization tasks. Section 5.5 presents experimental results of our framework on well accepted summarization data sets. Finally Section 5.6 concludes the chapter.

## 5.2 *weighted Hierarchical Archetypal Analysis*

We have already shown how AA and wAA are able to identify the extreme data points which are lying on the convex-hull of the given data set. In document summarization, given a 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. Alike, in the query-focused summarization, given a graph representation of a set of sentences, weighted by similarity to the given query, positively and/or negatively salient sentences are values on the weighted data set boundary. Weighted AA can be used to compute these extreme values, archetypes, and hence to estimate the importance of sentences in the target documents set. Unfortunately, matrices representing sentence similarity graphs can be often very complex. They can have complex inner structure, i.e. clusters of sentences representing different topics. Although AA and wAA are successful in treating a situations where data is convex and when the "outer" extreme values are of the interest, they have some clear limitations when their usage in summarization is considered. These concerns include 1) how can AA and wAA be used when data sets are complex, as it is with sentence similarity matrices, 2) and how to use AA and wAA for finding not only outer but also the inner extreme values.

### 5.2.1 *An illustrative example of wHAA*

In order to give a better grips, as an illustrative (already low-dimensional) example consider Figure 5.1 It depicts a typical "non-convex data" situation. We have drawn data points from 3 randomly positioned data clusters in 2D and were interested in computing the inner extreme values. By design, AA assigns clusters to the "extreme" data groups and not to the "inner" ones, as illustrated by Figure 5.1 (a). Although we can still reconstruct each data point perfectly, this is discouraging. The intrinsic (low dimensional) structure of the data is not captured and, in turn, the representation of the data is not as meaningful as it could be.

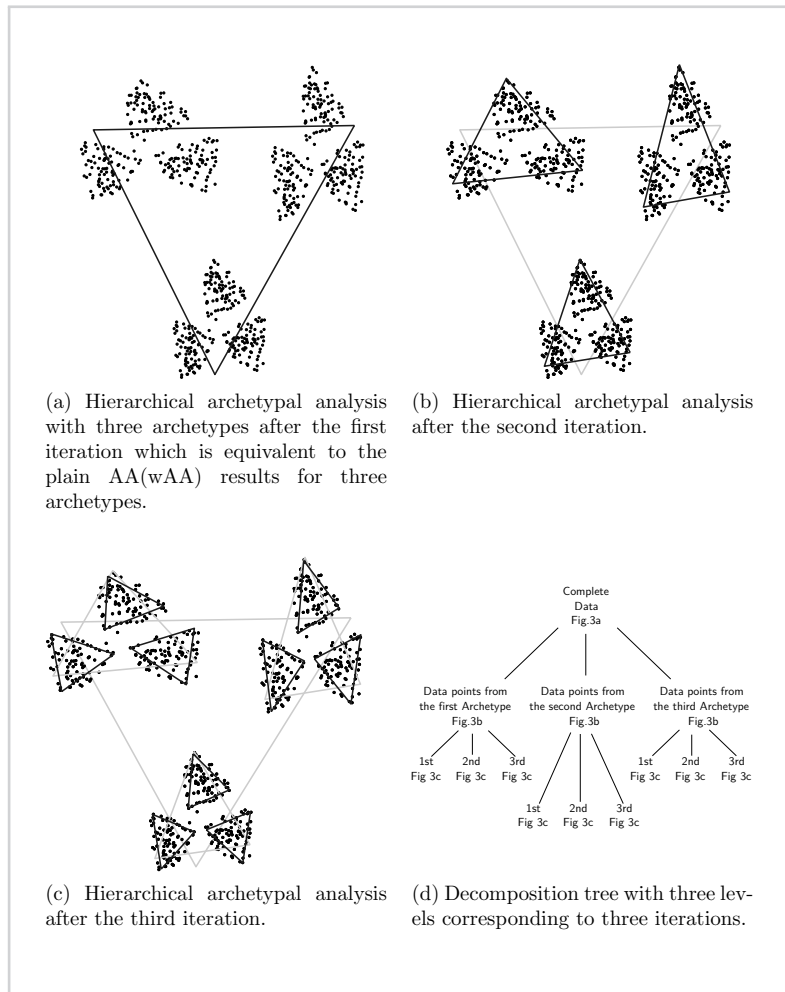


Figure 5.1

Didactic example of a set of data consisting of a randomly placed three clusters of data points. For archetype number  $z$  set to 3 in (a) are shown results from an application of the original version of AA. In (b) and (c) are intermediate results of the hierarchical version of AA. In (d) we present the tree of calls produced by applying wHAA for  $z = 3$  and the number of levels  $k = 3$ .

In this chapter as solution to mentioned limitations we propose hierarchical version of AA (wAA), namely wHAA. wHAA automatically adapts to the low intrinsic dimensionality of data as illustrated in Figures. 5.1(b,c,d). The algorithm design of the wHAA is based on the well known Divisive Analysis (DIANA). This variant of hierarchical clustering is also known as "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. wHAA is based on a hierarchical decomposition of  $\mathbb{R}^D$  in the form of a tree. Note that the standard wAA corresponds to a tree of depth zero.

Let us now first outline the proposed wHAA algorithm (as we implement and use in summarization) very generally. In subsequent subsections, we describe parts of the algorithm in more details and discuss different options for its usage in summarization.

### 5.2.2 General outline

Consider an  $n \times m$  matrix  $X$ . Given the number of archetypes  $z$  and the number of levels  $k$ , begin by running the wAA on the entire data set. Repeat the following steps at each level of the tree:

1. *Partitioning*: Assign each data point to only one partition based on the archetypal membership value. At the end of the step there should be  $z$  subsets of the parent data set.
2. *Processing*: run the wAA algorithm on each of the subsets.
3. *Ordering*: Order the new archetypes

Stop splitting the node when the number of levels in the tree is equal to  $k$  or when there are less data points than  $z$ . The final level is an ordered list of archetypes.

*Partitioning step*: Since the goal is to divide the data set to smaller subsets, in this step we use the wAA output from the previous iteration to cluster all data points to distinct archetypes. The wAA produces two stochastic matrices  $S$  and  $C$ . The latter one consists of rows representing archetypes and columns denoting the archetypal membership values of each data point. Therefore assigning each data point to one archetype (i.e. splitting the data set) is straightforward. The data point is assigned to only one archetype for which it has the highest membership value.

*Processing step*: In order to split data for the next iteration, we run wAA on each sub-dataset embedded in internal nodes. When stopping conditions are reached this step

is executed for the last time and the wAA results are embedded in the corresponding leaf.

### 5.3 Summarization method using wHAA

#### 5.3.1 Why weighted Hierarchical Archetype Analysis

The connection between the weighted archetypal analysis (consequently between the weighted hierarchical archetypal analysis) and the MDS can be easily identified. As discussed in Section 4.4 the weighted AA can be used in identifying the “best” summary sentences for the query focused summarization. The same reasoning applies to generic summarization task where we simply do not use the weight matrix, and ergo weighted AA becomes standard AA. Beside that AA (wAA) is generally able to select the “best” summary sentences, it has many other useful properties, including: (1) it is an unsupervised method; (2) 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; (3) the extracted sentence can be represented as a 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 (4) it readily offers soft clustering, i.e. simultaneous sentence clustering and ranking.

Unfortunately, despite all of its advantages, AA (wAA) lacks in a few fundamental aspects already described in previous section. There, as a solution, we proposed weighted Hierarchical Archetypal Analysis. Since the wHAA method basically inherits all the pros of AA (wAA) and introduces some new summarization-suitable properties, we use wHAA for the MDS task.

Let us now delve into the MDS task from the perspective of summarization algorithm based on wHAA, namely wHAASum.

#### 5.3.2 wHAASum algorithm

For a pool of sentences formed from the given document set, the problem is how to pick up the most representative sentences as a summary of this document set. The main idea of the method is simple: sentences are hierarchically soft-clustered into weighted 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

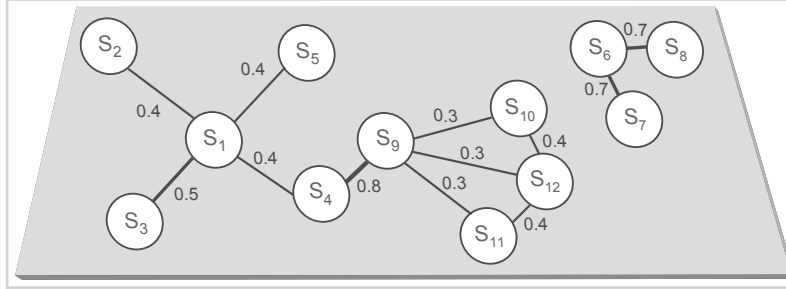
framework of the proposed method, wHAASum, consists of the following steps:

1. Construct the input matrix  $X$  depending in the summarization task.
2. Generate the input weight matrix  $W$ .
3. Perform wHAA on matrix  $X$  given the  $W$ .
  - i Build the tree of hierarchical decomposition where internal nodes contain splitting criteria and the leaves consist of the final decomposition matrices  $C_i$  and  $S_i$ .
4. Select  $l$  sentences to form the final summary.
  - i Start with the leftmost (the most significant) leaf and extract the sentence with the highest archetypal membership value. Continue with next leftmost leaf until the summary length constraint is met. That is, the sentence with the highest archetype membership value in each leaf's most significant archetype (row of matrix  $C_i$ ) is selected and if the summary length is not met, the extraction step continues with the next leaf, and so on. If the extraction has reached the last leftmost leaf but the required summary length is not met yet, the extraction continues with selection of the sentences with the second highest archetypal membership values, and so on.
  - ii Each selected sentence is compared to previously selected ones and if there is a significant similarity between them, i.e.  $\text{sim}(s_i, s_j) \geq 0.9$ , the newly selected sentence is not included in the summary.

In the above algorithm, the third and fourth steps are crucial. In the third step the goal is to generate the decomposition tree of the given matrix  $X$ . This is realized by employing the general algorithm given in Section 5.2.2. It is known that interior nodes of the produced tree will contain the splitting criteria based on wAAs archetypal clustering properties, embedded in matrices  $S_i$ . On the other hand, leaves will contain the final archetypal ranking given by matrices  $C_i$ . Since the algorithm from Section 5.2.2 orders the archetypes produced at each iteration in the decreasing order of significance, the most outstanding sentences are most likely to appear in leaves starting from the leftmost one.

Figure 5.2

Sentence similarity graph, where nodes denote sentences and weighted edges represent the similarity between corresponding sentences.



The fourth step in the above algorithm starts the sentence extraction from the leftmost leaf and follows with the next leftmost leaf until the desired summary length is reached. Worth to mention is the technique of actual sentence selection used at each leaf. Leaves contain  $S$  and  $C$  decomposition matrices. Since  $S$  matrices have been previously used in splitting, here  $C$  matrices can be used specifically for sentence ranking. Columns of these matrices denote sentences whereas rows represent ordered archetypes. Selecting sentence(s) from a  $C$  is therefore straightforward, the sentence with the highest membership value in the first row of the matrix is selected as the most outstanding one.

As already noted in previous chapter, in this way our method optimizes the two important aspects of the summarization, namely the relevance and the content coverage. The last important effect 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 (4.ii) of the above algorithm. In the following subsection we present the usage of wHAASum on an illustrative example.

### 5.3.3 An illustrative example

In order to demonstrate the advantages of wHAASum as the method for hierarchically simultaneous sentence clustering and ranking with respect to the sentence weights, a simple example from Figure 4.3 is for convenience here reproduced in Figure 5.2. Recall that we present the synthetic data set as an undirected sentence similarity graph, where nodes denote sentences and edges represent similarity between connected nodes.



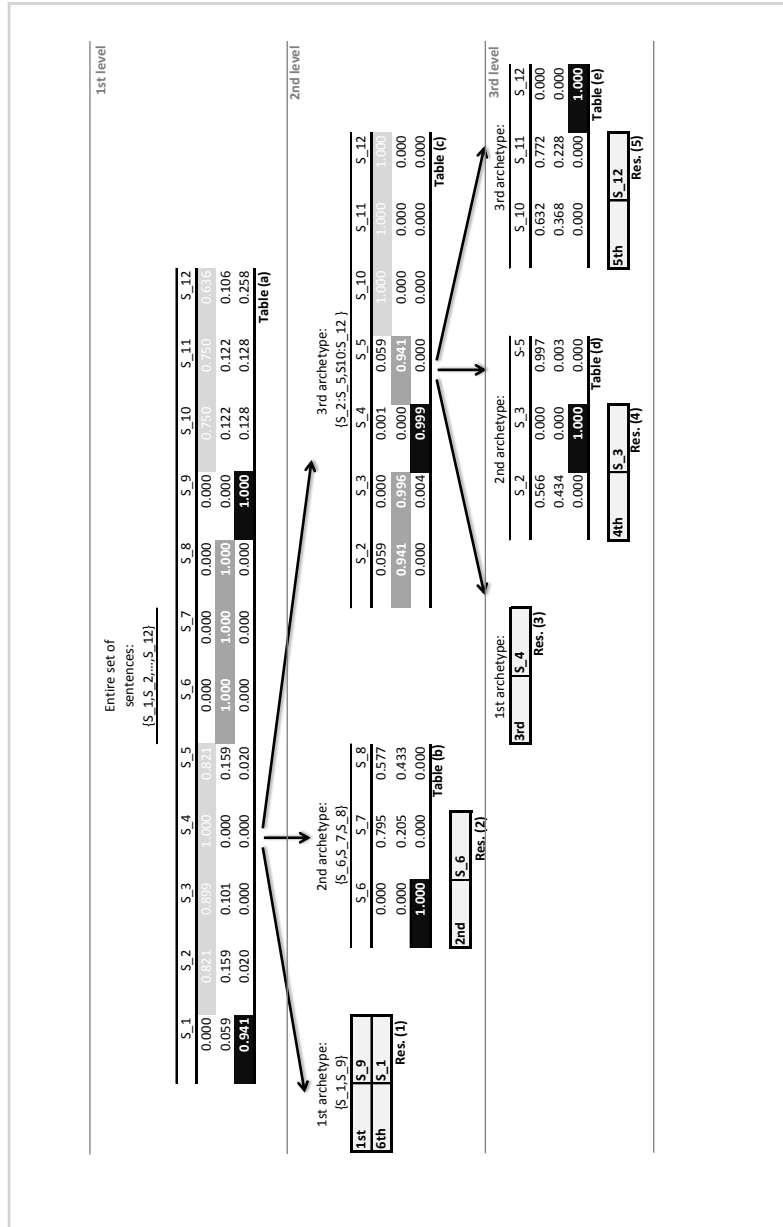


Figure 5.3

Results of Weighted Hierarchical Archetype Analysis on the illustrative example from Figure 5.2.



To better present the idea of wHAASum and even more simplify presentation in this example, we do not consider the sentence node weights (sentence to query similarity). This could be easily realized by connecting each sentence to the given query with the edge weighted by the corresponding sentence to query similarity.

By setting the number of archetypes  $z = 3$ , the number of levels  $k = 3$  and then by applying wHAASum we obtain the decomposition tree and the accompanied set of matrices  $S_i^T$ ,  $C_i^T$  and  $X^T C_i$ . In Figure 5.3.3, for the sake of simplicity, we only show the  $C_i^T$  matrices obtained by repeated use of wAA in hierarchical archetypal analysis. wHAA is like top-down clustering or divisive clustering. We start at the top with all sentences in one cluster. The cluster is then split using a flat wAA algorithm. This procedure is applied recursively until the number of sentences is less than the required archetype number  $z$ . As one can see from Figure 5.3.3, at the beginning all sentences are in one cluster. Then by applying wAA on the entire set of sentences three archetypes are obtained. Table (a) in Figure 5.3.3 represents matrix  $C^T$  produced after the first cycle of sequential wAA calls. Each row of this matrix is one of the estimated archetypes, and rows are presented in increasing order of the archetypal significance (the last row represents the most important archetype). Let us now further analyze this intermediate result. In fact, one can successfully extract summary sentences by plainly relying on this first result, as it is presented in [40]. The possible difficulty occurs when the plain wAA is trapped in a local minimum. It is obvious that  $s_9$  out of sentences  $\{s_9, s_1\}$  should be selected as a summary sentence since those have high archetypal membership values in the most significant archetype (3rd row). The local minimum issue in this problem is manifested when there is not enough information to make a right decision on sentence picking. A typical situation is reached at the second archetype (row) in Table (a). Here one can not easily decide which of the sentences  $s_6, s_7$  or  $s_8$  to pick. From the original graph it is obvious that sentence  $s_6$  should be picked. Similar issue appears in the first row of the same table. Here, the sentence  $s_4$  can be picked as the most significant, but still the sentence  $s_{12}$  is ranked lower, which does not reflect the reality. In our previous work (see Chapter 3) those issues were treated by increasing the number of archetypes. Here we continue the summarization process by further dividing the data set to sub archetypes. As described in previous sections, the complete set of sentences are now separated in three groups. Splitting is based on results from Table (a). The first group includes  $\{s_1, s_9\}$ , the second contains  $\{s_6, s_7, s_8\}$  and the last holds the rest of sentences, i.e.  $\{s_2, s_3, s_4, s_5, s_{10}, s_{11}, s_{12}\}$ . In the second level of Figure 5.3.3 three child nodes are

presented. Each one is in fact an archetype from the previous level, i.e. from the root. The Table Res(1) denotes the first leaf of the decomposition tree. From the previous level the third archetype, also the most significant one, has been used in forming the next leftmost subtree. But since the newly formed node has the number of elements (2) lower than the number of desired archetypes (3), the further splitting is stopped at this point and the node is converted to a leaf.

For interpretation, we identify the internal nodes as intermediate clustering results, leaves as final clustering and ranking results and the final archetypal sentences found in leaves as different types with different degree of potentially “good” and “bad” summary sentences. Then we set the observations in relation to them. Under this circumstances, the first produced leaf is the “best” sub-archetype containing ranking of the “best” archetypal sentences. The sentence  $s_9$  is the best of the best, which is followed by the sentence  $s_1$ .

The same procedure continues with other child-archetypes. Note that only the leftmost leaves are shown in Figure 5.3.3. This is due to logic of the original summarization algorithm, where the goal is to extract only the “best of the best” archetypal sentences. Table (b) from the second level is further clustered into three new child archetypes, but only the leftmost leaf is used for actual sentence selection. Table Res(2) shows the second selected sentence  $s_6$ . Note that this sentence could not be selected at the first level of the tree, however, at this point it is the “best” summary sentence of the second archetype. Table (c) represents the largest archetype produced by previous splitting and therefore it is further divided into three new child archetypes. Again, the leftmost leaf is used for direct sentence selection while the second and third archetypes are additionally processed.

This example shows that the output decomposition tree and matrices produced by wHAA describe the data structure well and at different levels of details.

#### 5.4 *The summarization framework*

Our proposed wHAA-based summarization framework can be directly used for different MDS tasks, including generic, query-focused, update and comparative summarization. To adapt the wHAASum to a different summarization task but still use the same method we model the summarization tasks differences by means of input matrix modeling. In this section, we formulate each summarization task by defining different matrix generation functions, henceforth labeling functions.

Table 5.1

A brief summary of labeling functions used in different summarization tasks. Remark: / indicates the absence of function

Summarization task	Labeling functions	
	Sentence similarity matrix	Weight matrix
General	$G_s = [sim_{norm}(s_i, s_j)]_{n \times n}$	/
Query-focused	$G_s = [sim_{norm}(s_i, s_j)]_{n \times n}$	$W = [sim_{norm}(q, s_i)]_{n \times n}$
Update	$G_s = [sim_{norm}(s_i, s_j)]_{n \times n}$ $U_s = [sim_{norm}(s_i, s_j)]_{n \times n}$	/ $W_1 = [1 - sim_{norm}(S_1, s_i)]_{n \times n}$ $W_2 = [sim_{norm}(q, s_i)]_{n \times n}$
Comparative	$C_s = \left[ (i, j) \mid \begin{cases} sim_{norm}(s_i, s_j) & \text{if } G(s_i) = G(s_j) \\ -sim_{norm}(s_i, s_j) & \text{if } G(s_i) \neq G(s_j) \end{cases} \right]_{i \times t}$	

Table 5.1 summarizes these labeling functions for different summarization tasks. The general procedure of methods for different summarization tasks is described in Section 5.3.2, while the only difference resides in the labeling functions.

The fundamental idea of labeling functions used in this work is based on representing the input sentences via sentence similarity graph. Note that the principal data structures used for graph representation are matrices, such as adjacency, incidence, etc. Therefore, the rest of this section describes the basic sentence similarity matrix generation process.

#### 5.4.1 General summarization

Given a set of documents, the general summarization task is to extract a set of sentences which can screen the overall understanding of the document set. Given the length limit to the summary, the generic summarization problem can be resolved by using the wHAA. In fact, for general summarization the algorithm wHAASum from Section 5.3.2 is used. Since there is no need for query incorporation, the labeling function is in its simplest possible form:

$$Gs = [sim_{norm}(s_i, s_j)]_{n \times n} \quad (5.1)$$

where  $sim_{norm}(\cdot)$  is the normalized similarity function for computing the similarity between sentences  $s_i$  and  $s_j$ . Resulting  $Gs \in \mathbb{R}^{n \times n}$  denotes a general sentence similarity matrix where  $n$  is the total number of sentences in document set  $D$ .

#### 5.4.2 Query-focused summarization

The query-focused multi-document summarization is a special case of generic multi-document summarization. Given a query, the task is to produce a summary which can respond to the information required by the query. Different from generic summarization which needs to preserve a typical semantic essence of the original document(s), the query-focused summarization purposely demands the most typical (archetypal) summary biased toward an explicit query.

Given a document set and a query  $q$ , we define the labeling functions as

$$Gs = [sim_{norm}(s_i, s_j)]_{n \times n} \quad (5.2)$$

$$W = [sim_{norm}(q, s_i)]_{n \times n} \quad (5.3)$$

where the first function represents the general information coverage, and the second function, the diagonal matrix  $W$ , represents the query-focused information coverage.

### 5.4.3 Update summarization

The multi-document update summarization was introduced by Document Understanding Conference (DUC) in 2007. It aims to produce a summary describing the majority of information content from a set of documents under the assumption that the user has already read a given set of earlier documents. This type of summarization has been proven extremely useful in tracing news stories: only new and update contents should be summarized if we already know something about the story.

Generally, this summarization task is based on the following scenario: Suppose that there is an open interest in a specific news topic and a need for tracking the related news as they emerge over time. In order to fulfill the information need of the users who are either overloaded with too many related news or are occasional readers of the given topic, the update summarization can provide summaries that only talk about what is new or different about this topic.

We formulate such a problem as follows:

Given a query  $q$  (representing the users interested topic) and two sets of documents  $D_1$  (previously read ones) and  $D_2$  (new ones), the update summarization aims to generate a summary of  $D_2$  related to the query  $q$ , given  $D_1$ .

First of all, the general summary of  $D_1$ , referred to as  $S_1$ , is generated by using the general summarization method presented in Section 5.4.1. Then, the update summary of  $D_2$  related to  $q$ , referred to as  $S_2$  is generated using the following labeling functions:

$$Us = [sim_{norm}(s_i, s_j)]_{n \times n} \quad (5.4)$$

$$W_1 = [1 - sim_{norm}(S_1, s_i)]_{n \times n} \quad (5.5)$$

$$W_2 = [sim_{norm}(q, s_i)]_{n \times n} \quad (5.6)$$

$$W = W_1 + W_2 \quad (5.7)$$

Here, the first function represents the general sentence similarity graph. The second function, diagonal matrix  $W_1$ , represents the dissimilarity between summary  $S_1$  and

sentences from document set  $D_2$ . Eq. (5.6) represents the query to sentence similarity diagonal matrix. Eq. (5.7) denotes the way of combining weight matrices  $W_1$  and  $W_2$ .

The main idea of  $S_2$  should be different from the main idea of  $S_1$ . This is ensured by weighting the archetypal analysis with dissimilarity of sentences to the first summary while producing the second summary. Since the normalized similarity is a value in range  $[0, 1]$ , its inverse is obtained by subtracting its value from 1. By penalizing the sentences similar to the first summary we reward the novel sentences and in this way attempt to model the update summarization problem. Also,  $S_2$  should cover all the aspects of the document set  $D_2$  as many as possible, which is again optimized using the general wHAA approach.

#### 5.4.4 Comparative summarization

In this section we investigate one of the recent summarization tasks, first proposed in [50] and referred to as Comparative Multi-document Document Summarization (CMDS).

CMDS is mainly about summarizing the diversities among related document groups. Formally, given a set of document groups, the comparative summarization is to produce a condense summary expressing the differences of these documents by extracting the most distinct sentences in each document group. While the goal of the classic document summarization is to extract the central information usually by taking into account the similarities among document collections, on contrary the aim of the comparative summarization is to capture differences among them.

We model the comparative summarization as follows: Extract the summaries  $S_1, S_2, \dots, S_N$  from the given  $N$  groups of documents  $G_1, G_2, \dots, G_N$ . Extracted summaries should be as divergent as possible from one another on the topic level while still expressing the central themes of corresponding groups.

We propose a different labeling function for the comparative summarization to generate the discriminant summary for each group of documents. The labeling function for the comparative summarization is defined as:

$$C_S = \left[ (i, j) \mid \begin{cases} sim_{norm}(s_i, s_j) & \text{if } G(s_i) = G(s_j) \\ -sim_{norm}(s_i, s_j) & \text{if } G(s_i) \neq G(s_j) \end{cases} \right]_{t \times t} \quad (5.8)$$

where  $G(s_i)$  is the document group containing sentence  $s_i$ ,  $sim_{norm}(s_i, s_j)$  is the normalized sentence similarity, and  $t = n_1 + n_2 + \dots + n_T$  is the total number of sentences

Table 5.2

Experimental data description.

	DUCo4	DUCo5	DUCo6	DUCo7	TACo8
Type of summarization	General	Query	Query	Update	Update
Cluster #	50	50	50	10	48
Documents # per set	10	25-50	25	25	10
Summary length	665 bytes	250 words	250 words	100 words	100 words

from all groups of documents.

Sentences from the same group are weighted by normalized similarity, mainly in order to grasp the centrality of the same. Sentences from different groups are weighted with an inverse normalized similarity. Since the normalized similarity is a value in range  $[0, 1]$ , the inverse is obtained by subtracting it from 0. By rewarding intra-group sentence similarity and by penalizing the inter-group sentences we attempt to model the Comparative Extractive Multi-document Document Summarization (CMDS).

### 5.5 Experiments

The experiments are conducted on four summarization tasks to evaluate our summarization framework. Results show that wHAASum outperforms many existing approaches.

The DUCo4 data set is used for the general (unrestricted) summarization task. As for the query-focused summarization, the DUCo5 and the DUCo6 data sets are used. The DUCo7 and the TACo8 data sets are used for the experiments on update summarization task. The compact view of the data sets can be found in Table 5.2. For the comparative summarization, we use the subset of the DUCo7 corpora to test our comparative summarization method.

All the tasks, except the comparative summarization, are evaluated by Recall-Oriented Understudy for Gisting Evaluation (ROUGE) evaluation package [43], which compares various summary results from several summarization methods with summaries generated by humans (see Eq. (3.11)). Here, we report the mean value over all topics



Table 5.3

General summarization task. Evaluation of the methods on the DUCo4 dataset. Remark: "-" indicates that the method does not officially report the results.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3242(10)	0.0641(10)	-
Best-Human	0.4182(1)	0.1050(1)	-
System-65	0.3822(3)	0.0921(3)	0.1332(2)
System-35	0.3708(4)	0.0834(6)	0.1273(5)
AASum-w2	0.3706(7)	0.0871(4)	0.1229(6)
LexRank	0.3784(5)	0.0857(5)	0.1312(3)
Centroid	0.3672(8)	0.0737(7)	0.1251(6)
LSA	0.3414(9)	0.0653(9)	0.1194(8)
NMF	0.3674(7)	0.0726(8)	0.1291(4)
wHAASum	0.4167(2)	0.0956(2)	0.1386(1)

of the recall scores of ROUGE-1, ROUGE-2, and ROUGE-SU<sub>4</sub> (skip-bigram plus unigram) [44]. For the comparative summarization, we provide some exemplar summaries produced by our summarization method. The detailed experimental results are described in the following.

### 5.5.1 Generic summarization

For the general summarization, we use DUCo4 as the experimental data. We observe through the experiment that the summary result generated by our method is the best when the archetype number and level number are set as  $z = k = 3$ . Consequently, we set  $z = k = 3$  when performing comparative experiments with other existing methods. We work with the following widely used or recent published methods for general summarization as the baseline systems to compare with our proposed method wHAASum: (1) BaseLine: the baseline method used in DUC2004; (2) Best-Human: the best human-summarizers performance (3) System-65: The best system-summarizer from DUC2004; (4) System-35: The second best system-summarizer from DUC2004; (5) AASum-W2: Archetypal analysis summarization system of the sentence similarity graph (6) Lex-PageRank: the method first constructs a sentence connectivity graph

Table 5.4

Query-focused summarization task. Evaluation of the methods on the DUC05 dataset.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU4
Avg-Human	0.4417(1)	0.1023(1)	0.1622(1)
Avg-DUC05	0.3434(9)	0.0602(9)	0.1148(9)
System-15	0.3751(6)	0.0725(6)	0.1316(6)
System-4	0.3748(7)	0.0685(7)	0.1277(7)
SNMF	0.3501(8)	0.0604(8)	0.1229(8)
NMF	0.3110(10)	0.0493(10)	0.1009(11)
LSA	0.3046(11)	0.0407(11)	0.1032(10)
PLSA	-	-	-
Biased-Lex	0.3861(4)	0.0753(4)	0.1363(5)
wAASum-W2	0.3790(5)	0.0735(5)	0.1365(4)
WHM	0.3906(3)	0.0817(2)	0.1393(3)
wHAASum	0.3948(2)	0.0808(3)	0.1435(2)

Table 5.5

Query-focused summarization task. Evaluation of the methods on the DUCo6 dataset.

Summarizers	ROUGE-1	ROUGE-2	ROUGE-SU4
Avg-Human	0.4576(1)	0.1149(1)	0.1706(1)
Avg-DUCo6	0.3795(10)	0.0754(10)	0.1321(10)
System-24	0.4102(5)	0.0951(3)	0.1546(4)
System-12	0.4049(7)	0.0899(6)	0.1476(7)
NMF	0.3237(11)	0.0549(11)	0.1061(11)
LSA	0.3307(12)	0.0502(12)	0.1022(12)
wAASum-W <sub>2</sub>	0.4075(6)	0.0872(7)	0.1531(5)
PLSA	0.4328(2)	0.0970(2)	0.1557(3)
Biased-Lex	0.3899(9)	0.0856(8)	0.1394(9)
SNMF	0.3955(8)	0.0854(9)	0.1398(8)
WHM	0.4212(4)	0.0921(5)	0.1517(6)
wHAASum	0.4245(3)	0.0924(4)	0.1578(2)

Table 5.6

Update summarization results. D/T Best and D/T Median stand for the DUC/TAC Best and Median results.

ROUGE	DUC <sub>07</sub>		TAC <sub>08A</sub>		TAC <sub>08B</sub>	
	2	SU <sub>4</sub>	2	SU <sub>4</sub>	2	SU <sub>4</sub>
D/T Best	0.1119(1)	0.1431(1)	0.1114(1)	0.1429(1)	0.1010(1)	0.1367(1)
D/T Median	0.0744(3)	0.1158(3)	0.0812(3)	0.1197(3)	0.0693(3)	0.1105(3)
wHAASum	0.0832(2)	0.1303(2)	0.0841(2)	0.1213(2)	0.0948(2)	0.1324(2)

based on the cosine similarity and then selects important sentences based on the concept of eigenvector centrality; (7) Centroid: the method extracts sentences based on the centroid value, the positional value and the first sentence overlap; (8) Latent Semantic Analysis (LSA): the method identifies semantically important sentences by conducting latent semantic analysis; (9) Non-negative Matrix Factorization (NMF): the method performs NMF on the sentence-term matrix and selects the high ranked sentences; (10) SNMF [27]: calculates sentence-sentence similarities by the sentence level semantic analysis, clusters the sentences via the symmetric nonnegative matrix factorization, and extracts the sentences based on the clustering result.

Table 5.3 shows the ROUGE scores of different methods using DUC<sub>04</sub>. The higher ROUGE score indicates the better summarization performance. The number in parentheses in each table slot shows the ranking of each method on a specific data set.

From the results showed, our method clearly outperforms the other rivals and is even better than the DUC<sub>04</sub> best team work.

### 5.5.2 Query-focused summarization

We conduct our query-focused summarization experiments on DUC<sub>05</sub> and DUC<sub>06</sub> data sets since the main task of both was the query-focused summarization. We compare our system with some effective, widely used and recently published systems: (1) Avg-Human: average human-summarizer performance on DUC<sub>2005/06</sub>; (2) Avg-DUC<sub>05/06</sub>: average system-summarizer performance on DUC<sub>2005/06</sub>; (3) System-15/24: The best system-summarizer from DUC<sub>2005/06</sub>; (4) System-4/12: The second best system-summarizer from DUC<sub>2005/06</sub>; (5) wAASum-W2: weighted Archetypal analysis summarization system of the sentence similarity graph; (6) Non-negative Ma-

trix Factorization (NMF): the method performs NMF on the sentence-term matrix and selects the high ranked sentences; (7) Latent Semantic Analysis (LSA): the method identifies semantically important sentences by conducting latent semantic analysis; (8) PLSA: employs the probabilistic latent semantic analysis approach to model documents as mixtures of topics; (9) Biased-LexRank: the method first constructs a sentence connectivity graph based on the cosine similarity and then selects important sentences biased toward the given query based on the concept of eigenvector centrality; (10) SNMF: calculates sentence-sentence similarities by the sentence level semantic analysis, clusters the sentences via the symmetric non-negative matrix factorization, and extracts the sentences based on the clustering result; (11) WHM [27]: document summarization is formalized as the optimization problem; the objective functions is defined as a weighted harmonic mean of the coverage and redundancy objectives;

The empirical results are reported in Tables 5.4 and 5.5. The results show that on DUCo5, our method outperforms the other systems; on DUCo6, our method achieves almost the best result. This is due to the novel adoption of the archetypal analysis, namely wHAA.

### 5.5.3 Update summarization

For update summarization we used the DUCo7 and TACo8 update task datasets.

The DUCo7 update task goal is to produce brief (100 words long) multi-document update summaries of newswire articles supposing that the user has already read a set of earlier articles. Each update summary should update the reader of new information about a particular topic. Given a DUC topic and its 3 document clusters:  $A$ ,  $B$  and  $C$ , the task is to create from the documents three short summaries such as: 1. A summary of documents in cluster  $A$ . 2. An update summary of documents in  $B$ , under the assumption that the reader has already read documents in  $A$ . 3. An update summary of documents in  $C$ , under the assumption that the reader has already read documents in  $A$  and  $B$ . Within a topic, the document clusters must be processed in chronological order; i.e., we cannot look at documents in cluster  $B$  or  $C$  when generating the summary for cluster  $A$ , and we cannot look at the documents in cluster  $C$  when generating the summary for cluster  $B$ . However, the documents within a cluster can be processed in any order.

The main task of TACo8 summarization track is composed of 48 topics and 20 news wire articles for each topic. 20 articles are grouped into two groups. The up-

Table 5.7

Results in comparative summarization: Sentences selected by our proposed wHAASum approach. The first column represents the cluster ID to which the selected sentences belong. Some unimportant words are skipped due to the space limit. The bold font is used to annotate the phrases that are highly related with the corresponding topic.

ID	Selected sentence
1	Stressing that the <i>introduction of a single currency</i> will be a <i>great contribution to the unity of an expanded European Union EU</i> , Juppe reiterated France's commitment to the <i>timetable and criteria of the single currency system</i> set in the Maastricht treaty, under which the <i>single European currency</i> , recently named <i>Euro</i> , will be <i>realized by January 1, 1999</i> .
2	They should pressure <i>Burma's military junta to enter a political dialogue</i> with Suu Kyi, Burma's ethnic nationalities, and the National League for Democracy, <i>the party elected overwhelmingly in 1990</i> but immediately denied its right to govern by the military.
3	<i>ETA, a Basque-language acronym for Basque Homeland and Freedom</i> , demands <i>the right to self determination</i> for Spain's three <i>Basque provinces</i> in the north. <i>ETA supported a Basque nationalist peace proposal ... negotiations between the government and ETA</i> , on issues like weapons and prisoners, ... <i>Basque country</i> .
4	France is running a near-record <i>12.4 percent jobless rate</i> with a <i>jobless population of 3.1 million</i> , one-third of which has been <i>unemployed</i> for more than a year. <i>Minister of Employment and Solidarity Martine Aubry</i> said ... that the government will present a <i>law on helping unemployed people ...</i> a series of measures will be taken in the fields of housing, health care and school.
5	Published in the October issue of the <i>American Journal of Public Health</i> , the other <i>study</i> examined the <i>health</i> and economic benefits of sustained moderate <i>weight loss</i> of 10 percent among persons who are <i>overweight or obese... Obesity</i> is not just a social stigma it is associated with <i>poor health</i> , from diabetes to heart ailments to gall bladder disease.
6	<i>The strength of Seinfeld</i> , along with the ratings garnered last year by the World Series and the Super Bowl, helped mask the problems for a time... Seeing that the <i>Jackie Chiles</i> character is the <i>only one to resurface since the comedy ended</i> , Morris was merely pointing out the irony of it all ...

date summarization task is to produce two summaries, using the initial summary (TACo8A), which is the standard query focused summarization, and the update summary (TACo8B) under the assumption that the reader has already read the first 10 documents.

Table 5.6 shows the comparative experimental results on the update summarization. In Table 5.6, “DUC/TAC Best” and “DUC/TAC Median” represent the best and median results from the participants of the DUC2007 and TAC2008 summarization tracks.

As seen from the results, the ROUGE scores of our method are higher than the median results. The good results of the best team usually come from the fact that they employ advanced natural language processing techniques. Although we can also utilize those kind of techniques in our solution, our goal here is to demonstrate the effectiveness of formalizing the update summarization problem using the weighted hierarchical archetypel analysis and therefore *we do not use any advanced NLP technique*. Experimental results demonstrate that our simple update summarization method merely based on the wHAA can result in competitive performance for the update summarization.

#### 5.5.4 Comparative summarization

For the comparative summarization task, we use randomly selected 6 clusters of documents from the DUCo7 corpora to generate comparative summaries using the wHAA-Sum summarization method. The data set contains 6 clusters as follows: 1. Steps toward introduction of the Euro; 2. Burma government change 1988; 3. Basque separatist movement 1996-2000. 4. Unemployment in France in the 1990s; 5. Obesity in the United States and possible causes for US obesity; 6. After “Seinfeld” TV series;

Looking at the results by our wHAASum sentence selection method in Table 5.7, each of the sentences represents one cluster respectively and summarizes well specific topics of each cluster. In Table 5.7, we also highlight some keywords representing the unique features of each topic. Note that the sentences extracted by wHAASum for each topic are not just discriminative but they also present the essence of the topic. For example, the summary of topic 3 defines the acronym ETA and clearly explains their demands. Another complex example is the summary of topic 6 where we are interested in what became of the cast and others related to the “Seinfeld” TV series after it ended. Again, the selected summary sentence answers well the concerns given

by the query and at the same time it is completely dissimilar to the summaries of other topics. Note also how successfully the summary No.1 defines and explains the issues of introducing the Euro.

### *5.6 Conclusion and future work*

In this chapter, we present a novel summarization framework based on weighted hierarchical archetypal analysis, namely wHAASum. We used the weighted Hierarchical Archetypal Analysis to select “the best of the best” summary sentences by producing the hierarchical decomposition in the form of a tree. All known summarization tasks, including generic, query-focused, updated, comparative summarization can be treated by this framework. The empirical results show that this framework outperforms the other related methods in generic summarization and that it is competitive in other summarization tasks. The skill to address these summarization problems is built on the weighted hierarchical archetypal analysis problem itself, and on various input matrix modeling functions for corresponding summarization tasks.

Our future work resides still in the area of summarization, but we would like to widen the area of wHAASum’s usability by exploring the possibilities of utilizing it in other fields such as opinion and biomedical summarization.



## *Final Discussion*

## 6.1 Complexity Analysis

In this section we present the complexity analysis of the proposed summarization methods. We first present the theoretical time and space complexity for the preprocessing step. Then we continue with the time complexity analysis for archetypal analysis itself. And finally, we conclude the section with theoretical and empirical complexity analysis of the summarization methods proposed in the earlier sections.

### 6.1.1 Preprocessing

As it was presented earlier, in order to obtain the sentences similarity graph one needs to compute the similarity values for all possible pairs of sentences in order to connect them in the sentence similarity graph. We used the vector space model to represent sentences from given documents. The vector space model is an algebraic model for representing sentences as vectors of terms. Computing the similarity of sentences then reduces to computing the cosine similarity. The cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.

Assuming that multiplication and addition are constant-time operations, the time complexity of computing the cosine similarity where  $m$  is the biggest number of terms is therefore  $O(m) + O(m) = O(m)$ . The only auxiliary space we require during the computation is to hold the 'partial dot-product so far' and the last product computed. Assuming we can hold both values in constant-space, the space complexity is therefore  $O(1) + O(1) = O(1)$ . But since we need to compute the sentence similarity for every pair of sentences then the time and space complexity of generating the sentence similarity graph becomes  $O(n(n-1)/2)$ , here  $n$  is the number of sentences. Even though a quadratic complexity can be seen as problematic, given that typical number of sentences in summarization is not too big, then the overall time consumption is on an acceptable level. In order to give better grips on the time complexities in Table 6.1 we give some empirical measurements.

### 6.1.2 Archetypal Analysis

The convex hull or envelope of a data matrix  $X$  is the minimal convex set containing  $X$ . While the problem of finding the convex hull is solvable in linear time (i.e.,  $O(n)$ ) [51] the size of the convex set increases dramatically with the dimensionality of the

data. The expected size of the convex set for  $n$  points in general position in  $m$  dimensional space grows with dimension as  $O(\log(n^{m-1}))$  [52]. In the paper of Cutler and Breiman [33] the model was estimated by non-negative least squares such that the linear constraints were enforced by introducing quadratic penalty terms in the alternating updates of  $S$  and  $C$ . Alternatively, the standard non-negative quadratic programming solvers with linear constraints can be invoked [36] for each alternating sub problem solving  $S$  for fixed  $C$  and vice versa. We found however, that the following projected gradient method inspired by the projected gradient method for NMF [53] using the normalization invariance approach worked efficiently in practice. By using the later approach, the overall computational complexity is the same for AA as for NMF [53]. When only considering  $T$  candidate points used to define the archetypes as proposed in Bauckhage and Thureau [36] this latter complexity can be further reduced to  $O(zmn)$ , where  $z$  is the number of archetypes,  $m$  the dimensionality of the feature space and  $n$  the number of observations.

### 6.1.3 Archetypal analysis based document summarization

Given that AA based document summarization system consists of preprocessing and processing steps then the overall time complexity can be found by summing the latter two complexities, i.e:  $O(O(n(n-1)/2)+O(zmn))$ , where  $n$  is the number of sentences,  $m$  is the number of terms, and  $z$  is the number of archetypes.

Although the overall time complexity seems to be quadratic, since the number of archetypes  $z$  is usually very low (maximum 16 in our case) and the number of sentences and therefore number of terms are bounded to the length of input documents the overall time spent for producing the summary is time affordable. In order to give a better gist of the time complexity we report the elapsed time in producing the summary for different document(s) lengths in in Table 6.1. The time elapsed in computing the summaries are measured on processor with following specifications: Intel(R) Core(TM) i5 CPU M 450 @ 2.45GHz with 4GB RAM memory. The first two columns (document(s) length in KB, and the total number of sentences) represent the input values, while the rest three columns (pre-processing, archetypal analysis based sentence selection and total time spent in seconds) are the measured times. Overall quadratic time complexity can be noticed also in the experimental results, but one should bear in mind that this methods are intended to be used in summarizing the news articles. The document lengths in real situations are not expected to be more than 100KB.

Table 6.1

Observed execution time spent on calculations (in seconds).

Document(s) length	# of sentences	Pre- processing	AA	Overall
1KB	13	0.09	1.73	1.83
20KB	160	1.40	1.11	2.52
45KB	366	2.28	4.82	7.10
100KB	744	5.89	9.21	15.11
644KB	5112	191.06	207.89	398.95

## 6.2 Limitations

In this section we describe some limitations and ways how to overcome them. We first present the well-known limitations of the existing automatic evaluation method. Then, we continue with the guided summarization task, the most recently proposed summarization task and the relation of our methods to it. Then, we conclude the section by discussing the length of the selected sentences and its impact on summarization quality.

### 6.2.1 Evaluation

The time consumption, cost, and human subjectivity are only few of the issues of the human evaluation of text summarization results. As the alternative to human evaluation more efficient and objective automatic evaluation methods have been developed. The most widely used method ROUGE [43], which is also used in our work, uses lexical N-grams to compare the human written summaries with the computer generated summaries.

The fundamental problem with this kind of automatic summary evaluation method is that it only relays on the surface level formulation, and that it is not sensitive to syntactic structure.

To overcome these types of shortcomings, the Basic Element summarization method was developed and tested in 2006 [54]. This method facilitates matching of expressive variants of syntactically wellformed units called Basic Elements (BEs). The system

achieved fairly good correlation with human evaluation.

A new implementation of the BE method, called BE with Transformations for Evaluation (BEwTE), that includes a significantly improved matching capability using a variety of operations to transform and match BEs in various ways is described in [55]. The extended BE method generally performs well against other automated methods for evaluating summaries. The intuition behind Basic Elements is to decompose summaries to lists of minimal length syntactically well-defined units (BEs) and then to compare the two lists to obtain a similarity score. In order to extract the BEs, they first parse the summaries using the Charniak parser [56], identify named entities using the LingPipe NER system, and then extract the BEs using a series of Tregex rules. If a token identified for extraction by a BE extraction rule falls within a string recognized by a Named Entity Recognition (NER) system as an entity, the entire named entity string is extracted in place of the word. Then the series of transformations occurs during a step between BE extraction and the overall score computation. For the sake of illustration and without diving into details we are giving only the list of transformations used in this step: Add/Drop Periods, Noun Swapping for ISA type rules, Prenominal Noun Prepositional Phrase, Nominalization, Denominalization, "Role" Transform, Adjective to Adverb, Adverb to Adjective, Pronoun Transform, Name Shortener/Expander, Abbreviations/Acronyms, Lemmatization/Delemmatization, Synonyms, Hypernym/Hyponym, Pertainyms Transform, Membership Meronym/Holonym Transform, Preposition Generalization.

Since the original version of ROUGE is most widely used method for summary evaluation, we followed the conventional evaluation framework and used it in all our previous experiments. But in order to present an alternative to standard evaluation scheme and to show how our summarization method performs when it is evaluated with a novel evaluation approach we designed a new set of experiments. In them we evaluate our method by using the BEwTE. Table 6.2 shows the BEwTE scores of the best and average human, the first and the second best automatic system, and the wHAASum system summarizer performances. The higher BEwTE score indicates the better summarization performance. The numbers in parenthesis in each table slot show the ranking of each method on a specific data set. In this experiment we compare our methods performance on data sets from DUCo4 to DUCo7. From the results showed, our method is among the best automatic summarization system for each year. Oscillations on the top few positions of our method can be explained by the fact that our

Table 6.2

Comparison of BEwTE scores.

	DUCo4	DUCo5	DUCo6	DUCo7
Best-Human	0.2563(1)	0.2755(1)	0.2845(1)	0.3213(1)
Avarage-Human	0.2352(2)	0.2502(2)	0.2399(2)	0.2681(2)
1st Best-System	0.2217(3)	0.1625(4)	0.1768(3)	0.2193(3)
2nd Best-System	0.1983(5)	0.1595(5)	0.1696(5)	0.2138(4)
wHAASum	0.2162(4)	0.1646(3)	0.1743(4)	0.2115(5)

method is merely statistical, while, on the other side, the data sets, the summarization tasks and the competing methods have become more semantic oriented from year to year. Nevertheless, given all limitations and obstacles our method still performs very well.

### 6.2.2 Guided Summarization

Since the idea of automatic summarization has been first introduced [1] there was one hidden problem in the very nature of the task formulation. The absence of a single "gold standard" that automatic systems could model was and still is one of the main problems in automatic text summarization. Summarization is generally based on an ambiguous notion of "importance" of information mentioned in the original text. In fact, methods for automatic extractive summarization have been mainly promoted to being able to capture information around this concept of "importance". But it is notably subjective and content-dependent. Another dimension of the problem is that this "weak" formulation has opened the door for introducing various methods for summarization generally known as statistical extractive methods, and even for automatic summary evaluation methods. Those methods that select high-scoring sentences based on term frequency provide a good baseline for summarization, but they are blind to synonyms and equivalent expressions in the source text and, in multi-document summarization, they can result in high degrees of redundancy and non-readability.

A second problem is using completely extractive methods. Recently an experiment on human extractive summarization [57] has showed that even the human summarizers as the examples of the best content-selection mechanism are unable to create good

summaries if they are limited to putting together sentences taken out of context from a number of independently written articles.

The guided summarization task is specifically developed and presented as the novel summarization task which aims to address those issues. The guided summarization task describes a precise, solid information model that automatic summarization systems can emulate. This is treated by defining topics that fall into template-like categories and contain highly predictable elements, as well as explicitly guiding the creation of human reference summaries to contain all these elements. On the other side, in order to promote the abstractive summarization, the guided summarization emphasizes the use of information extraction techniques and other meaning-oriented methods, and thus encourages a move towards abstractive summarization by using the sub-sentential level analysis.

The guided summarization task is to write a 100-word summary of a set of 10 newswire articles for a given topic, where the topic falls into a predefined category. There are five topic categories: Accidents and Natural Disasters, Attacks, Health and Safety, Endangered Resources, Investigations and Trials. Participants (and human summarizers) are given a list of important aspects for each category, and a summary must cover all these aspects (if the information can be found in the documents). The summaries may also contain other information relevant to the topic.

Given this new summarization task and a requirement to put things into a more realistic perspective, in the next subsection we give some insights regarding the ways for adapting the presented approaches to the direction of guided summarization.

#### *Motivation for using semantic role graphs*

Since humans tend to include sentences containing most frequent words in their summaries, the word-based frequency calculations for sentence scoring are base-born approaches for MDS. However, this approach is semantically incomplete, since words alone usually do not carry semantic information. On the other hand, even if humans do not always agree on the content to be added to a summary, they perform very well on this task. Therefore our goal should be to find a way of mimicking the cognition behind the human like summarization process. Since the abstractive summarization is a hard task and the extractive is relatively easier we can propose a compromise. One can identify six editing operations in human abstracting: (i) sentence reduction; (ii) sentence combination; (iii) syntactic transformation; (iv) lexical paraphrasing; (v) gen-

eralization and specification; and (vi) reordering. Summaries produced in this way approximate the human summarization process more than the extraction does. One of possible methods of using the graph based model in the abstractive summarization setting comes with the idea of using a part of sentences instead of the whole sentences as the textual units in creation of the graph model. Some of the above mentioned operations can be redefined in terms of the graph operations and as a result establish the bases for moving toward, in the worst case, the pseudo-abstractive summarization. Our motivation for using SRL (semantic-role labeling) frames in sentence scoring for MDS originates from given concerns. Instead of using individual terms for sentence scoring, we exploit semantic arguments and relations between them by using the psychology cognitive situation model, namely the Event-Indexing model.

#### *Event Indexing Model and Semantic Role Labeler*

According to the Event-indexing model a human-like system should keep track of five indices while reading the document. Those indices are protagonist, temporality, spatiality, causality and intention, with the given descending order of importance. One can also show that the semantic role parser's output can be mapped to the above proposed cognitive model. Semantic roles are defined as the relationships between syntactic constituents and the predicates. Most sentence components have semantic connections with the predicate, carrying answers to the questions such as who, what, when, where etc. From the aspect of the semantic parser, frame arguments can be mapped to cognitive model indices as follows: A protagonist can be found in an answer to question "who", or more precisely in arguments  $A_0$  or  $A_1$  or  $A_2$ . Argument  $A_0$  is the subject of the frame, as shown in Table 6.3,  $A_1$  is the object and  $A_2$  is the indirect object. Although in original work the protagonist is defined as a person around whom the story takes place, we see it reasonable to expand the notion of protagonist to the subject or object that can be everything, from a person to an organization or some abstract concept. Temporality is the temporal information in each frame and can be extracted from the frame argument  $AM_{TMP}$ . Spatiality is the space or location information of each frame and is equal to argument  $AM_{LOC}$ . Causality indexing is concerned with actions of frames so it can be mapped to the frame predicate. The intentionality-indexing is quite vague but since its weight of significance is less than of the others, as defined in the original work, we decided to omit it in this early versions of the system. The SRL parser takes each sentence in the document set and properly



Table 6.3

Representation of the label arguments and modifiers.

Label	Modifier	Label	Modifier
rel	verb	$AM_{ADV}$	Adverb mod
Ao	Subject	$AM_{DIR}$	Direction
A1	Object	$AM_{DIS}$	Discourse mark
A2	Ind. object	$AM_{LOC}$	Location
A3	Start point	$AM_{MNR}$	Manner
A4	End point	$AM_{NEG}$	Negation
A5	Direction	$AM_{PRD}$	Sec. Predicate
		$AM_{PRP}$	Purpose
		$AM_{TMP}$	Temporal mark

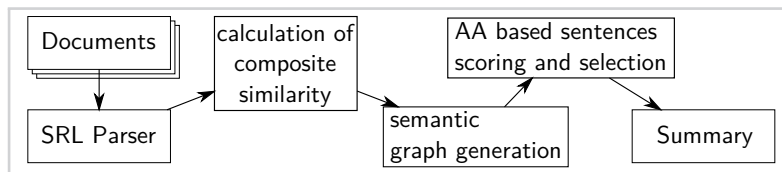


Figure 6.1

Archetypal analysis based guided summarization system prototype.

labels the semantic word phrases. We refer to these phrases as semantic arguments or shortly arguments. There is an issue related to the SRL parsing process that we should take into account. For each verb in a sentence, the SRL parser provides a different frame. It considers the verb as the predicate of the sentence and tries to label the remaining part of the sentence as proper arguments. However, if the selected verb is not the actual predicate, the parser fails to identify most of the words as a part of an argument. Therefore, we consider the frame that leaves the least number of terms unlabeled as the complete parse of the sentence. In our calculations we use also the rest of frames but we treat them as incomplete. Since we don't want to lose information that can be brought to the resulting graph, instead of eliminating partially parsed frames we use them, but with lower weight in the similarity calculation.

Table 6.4

The average number of sentences and the average sentence length in words in summaries produced by top ten human, system and AA based summarization systems.

summarizers	Average number of sentences in a summary		Average sentence length in words	
	DUC2005	DUC2006	DUC2005	DUC2006
human	11.7	14.5	20.2	16.0
system	7.3	8.5	30.7	40.0
AA based	7.0	7.3	42.2	38.1

### *Archetypal analysis based guided summarization system prototype*

The guided summarization method based on Archetypal analysis could work in the following way, as illustrated in Fig 6.1. First, the documents are given to the SRL parser where the semantic arguments from each parsed sentence are extracted. Then the composite similarity between all semantic frames based on the event-indexing model is calculated. Next a semantic graph is generated, where nodes represent semantic frames and edges denote the composite similarity values. Given the semantic graph one can easily adapt one of the archetypal analysis based sentence ranking methods presented in earlier section for sub-sentence parts ranking and selection. Subsequently, the top scoring sub-sentential entities can be selected one-by-one and put into the summary.

### *6.2.3 The length of the selected sentences and its impact on summarization quality*

Most multi-document summarization methods utilize some sentence related features to calculate the sentence significance. However, to the best of our knowledge no empirical studies have been performed to determinate the impact or even if it exist the contribution made by sentence length information. In this subsection, we focus on the sentence length, investigating (1) the association between the lengths of the selected sentences and the quality of the summary produced by combining these sentences; (2) the tendency of the AA based summarization methods for selecting the longer sentences. In order to study how sentence length influences the performance of the summarization systems, we used datasets for the multi-document summarization task from the DUC2005 and the DUC2006. For each set, DUC provides few,

*Table 6.5*

Pearson's  $r$  values for the linear correlation between the length of the selected sentences and the quality of produced summaries

	DUC2005	DUC2006
Human-summarizers	-0.4143	-0.6098
System-summarizers	-0.6700	-0.8688
AA based summarizers	-0.8697	-0.8418

usually four, human-written summaries and the submissions of all participating automatic summarizers. Table 6.4 shows the average number of sentences and the average sentence length in words in summaries produced by the top ten human, system and AA based summarization systems. These figures suggest few things: 1) the human summarizers tend to produce larger number of shorter sentences, 2) the top performing system summarizers produce the summaries by extracting fewer number of longer sentences, and 3) similarly to the latter ones, the AA based summarizing systems we proposed in the thesis are producing summaries containing fewer number of longer sentences. In order to better illustrate the relation of the length of the selected sentences and the quality of produced summaries in Table 6.5 we report the Pearson's  $r$  values for the linear correlation between these two variables. Results clearly suggest that there is a strong negative correlation between these two variables, which can be read as follows - as much as the average sentence length of the selected summary sentences increases the summary quality decreases. From these experiments, it appears that directly modeling the sentence length in a summary can be effective on increasing its quality. On the other hand this appears to be the hard problem since many automatic extractive summarization systems utilize some kind of term frequency related features to determine the sentence importance. These systems therefore tend to select the longer sentences containing the higher number of significant facts. In the context of archetypal analysis based summarization systems as a part of future work one can investigate the approaches for optimizing the sentence length along the other parameters.

Table 6.6

Evaluation of the AA based summarization methods on the TAC2008 and the TAC2009 datasets.

Task	method	TAC2008		TAC2009	
		ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
General	AASum	0.4065	0.0928	0.4050	0.1026
Query	wAASum	0.4271	0.0956	0.4336	0.1077
Update	wHAASum	0.3644	0.0762	0.3797	0.0828

#### 6.2.4 Evaluation of the AA based summarization methods on some newer data sets

In this subsection we present the performance test results for our summarization methods on some newer datasets such as the TAC datasets from 2008 and 2009. The goal of the TAC Summarization track is to foster research on systems that produce summaries of documents. The focus is on systems that can produce well-organized, fluent summaries of text. Piloted in DUC 2007, the TAC 2008 and TAC 2009 Update Summarization task is to generate short (100 words) fluent multi-document summaries of news articles under the assumption that the user has already read a set of earlier articles. Since 2010, TAC has been encouraging only the guided summarization task. The Guided Summarization task aims to encourage summarization systems to make a deeper linguistic (semantic) analysis of the source documents instead of relying only on document word frequencies to select important concepts, which was not our intention in this work.

Having in mind that the AASum is a general, the wAASum is a query oriented and the wHAASum is a generic summarization method we decided to adapt the TAC 2008 and TAC 2009 datasets to our needs as follows: (1) to evaluate the AASum performance on the new datasets we treat them as the general document summarization test sets by ignoring the query and update information; (2) to examine the wAASum we adapt the datasets to the query oriented test set by simply ignoring only the update dimension of the problem; (3) and finally, in order to test the wHAASum we used the datasets as they are. Tables 6.6 and 6.7 show the comparative experimental results where the “TAC Best” and the “TAC Median” represent the best and the median systems results from the participants of the TAC2008 and TAC2009 summarization

Table 6.7

Comparison of the ROUGE scores of the wHAASum with the TAC Human and System summarizers on the TAC2008 and TAC2009 datasets.

	TAC2008		TAC2009	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
TAC Human Best	0.4328(1)	0.1320(1)	0.4389(1)	0.1371(1)
TAC System Best	0.3730(2)	0.1040(2)	0.3831(2)	0.1130(2)
TAC System Median	0.3433(4)	0.0752(4)	0.3410(4)	0.0796(4)
wHAASum	0.3644(3)	0.0762(3)	0.3797(3)	0.0828(3)

tracks. The input parameters are set to their default values defined previously in the text, where the number of archetypes is set to  $z = 16$  for the AASum and wHAASum, the archetype number and the level number are set as  $z = k = 3$  for the wHAASum. Results from these tables suggest few things: 1) the high results in the query and the general summarization tasks are achieved in view of the fact that we purposely ignored the update aspect of the original datasets and we fused the former and the latter document versions in one dataset 2) the results of the wHAASum in update summarization are much better than the TAC system average and are almost as good as the best TAC system summarizers.



## *Conclusion and Future Work*

7

## 7.1 Conclusion

In this section we briefly summarize the principal scientific contributions. With each contribution we list the dissertation sections where the topic is discussed. In addition we also list the references to our publications that discuss the topic.

- *Archetypal analysis for multi-document summarization.* The main contributions of Chapter 3 are the following: We presented 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 the document collection. Document summarization was formalized as the Archetypal Analysis problem that takes into account relevance, information coverage, diversity and the length limit. The chapter also showed how AA can be used for simultaneously sentence clustering and ranking. The chapter 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. It was also showed that AASum is an effective summarization method. Experimental results on the DUC2004 and DUC2006 datasets demonstrated the effectiveness of the proposed approach, which compared well to most of the existing matrix decomposition methods in the literature. The content and drafts of the methods presented in Chapter 3 were published in [40].
- *Weighted archetypal analysis for query-focused multi-document summarization.* The main contributions of Chapter 4 are the following: The chapter has formalized the problem of query-focused document summarization as the weighted archetypal analysis problem. Additionally, it has presented our study of how to incorporate query information in the own nature of AA and how to use the weighted version of AA for simultaneous sentence clustering and ranking. We have examined the proposed method on several input matrix modeling configurations, where the chapter reports the best results on the multi-element graph model. The presented work has proven that wAASum is an effective summarization method. Experimental results on the DUC2005 and DUC2006 datasets demonstrated the effectiveness of the proposed approach, which compared well to most of the existing matrix factorization methods in the literature. The content of the chapter is partially based on our work [45].



- *Weighted hierarchical archetypal analysis based generic multi-document summarization framework.* The main contributions of Chapter 5 are the following: The chapter presents a novel summarization framework based on weighted hierarchical archetypal analysis, namely wHAASum. We used the weighted Hierarchical Archetypal Analysis to select “the best of the best” summary sentences by producing the hierarchical decomposition in the form of a tree. All known summarization tasks, including generic, query-focused, updated, and comparative summarization can be treated by this framework. The empirical results showed that this framework outperforms the other related methods in generic summarization and that it is competitive in other summarization tasks. The skill to address these summarization problems is built on weighted hierarchical archetypal analysis problem itself, and on various input matrix modeling functions for corresponding summarization tasks. The content of the chapter is partially based on our work [58].

## 7.2 Future Work

We have already discussed possible directions for future improvements of methods in specific application domains, namely for general summarization (Section 3.4), query-oriented summarization (Section 4.6), and in generic summarization framework (Section 5.6). We now take a broader view on prospective future work which goes beyond the text summarization applications that are the topic of this thesis.

*Parameters selection.* Although parameter selection doesn't play a central role yet it has an important function in our research, especially in hierarchical weighted archetypal analysis. The main idea of parameter selection is to choose a subset of relevant parameters for building robust archetypal analysis based summarization models. In our current research we use standard ad hoc parameter selection techniques. Since the most important parameter in our methods is the number of archetypes, we have mainly projected the general parameter selection problem to choosing the best value for the number of archetypes to be computed. For the plain and the weighted archetypal analysis we used a simple approach for choosing the value of the number of archetypes: we run the algorithm for different numbers of the archetypes where the selection criteria was the maximization of the summary evaluation outcome. In this way we found some values for which we believe are the most adequate. Similarly, in order to define

the two most important parameters in hierarchical weighted archetypal analysis, we followed the same direction of thoughts, where we decided on some ad hoc values for the archetype number and the number of levels on decomposition tree. Yet, at the end one can still argue our decisions and arguments, hence we believe that a deeper, more comprehensive, and accurate analysis on parameter selection and tuning with even more detailed experiments should be done as the part of future work.

*Application of the developed methods on Guided summarization.* Given the new task of guided summarization and requirement to put things into a more realistic perspective in Subsection 6.2.2, we presented some insights on the ways for adapting the archetypal analysis based summarization approaches to the direction of guided summarization. Concrete implementation and evaluation is purposely left for the future work.



# *Razširjeni povzetek*

*A*

## A.1 Uvod

Glavni cilj te disertacije je razviti novo metodo za povzemanje besedil, ki izkorišča prilagodljivost modeliranja z uporabo grafov in učinkovitost analize z arhetipi.

Minilo je že pol stoletja od Luhnovega pionirskega članka o samodejnem povzemanju besedil [1]. V tem času je povzemanje besedil postajalo vedno bolj pomembno in objavljenih je bilo že mnogo člankov. Danes je svetovni splet sestavljen iz milijard različnih dokumentov, večinoma besedil, in še vedno narašča eksponentno. To je sprožilo zanimanje v razvoj metod za samodejno povzemanje besedil. Takšne metode so bile sprva zasnovane tako, da so iz besedila ali gruče besedil izluščile kratek in naraven povzetek najbolj pomembnih informacij. Pred kratkim pa so se začele pojavljati nove naloge povzemanja besedil in ustrezne rešitve, ki so, čeprav še niso dozorele, že dokazale svojo uporabnost.

Ekstraktivni povzetki (angl. extractive summary) so povzetki sestavljeni iz stavkov, ki se nahajajo v besedilih, ki jih povzemamo. Izvlečki (angl. abstractive summary) so povzetki, ki razkrivajo bistvo izvornih besedil, praviloma z besedami avtorja besedil, čeprav so lahko stavki tudi spremenjeni. Prva naloga zgodnjih del na temo povzemanja besedil je bila povzemanje posameznih besedil. Z razvojem raziskav se je pojavila nova naloga povzemanja - povzemanje skupin besedil (multi-document summarization). Glavna motivacija za MDS so primeri s svetovnega spleta, kjer se srečujemo z velikim številom odvečnih besedil z isto tematiko. Pri prvih uporabah sistemov za povzemanje spletnih besedil so MDS uporabili na skupinah novic o istem dogodku, da bi ustvarili povzetke za strani za brskanje po spletu [2]. Povzetke lahko določimo tudi z njihovo poanto. Povzetku, ki bralca seznanja s temo izvornih besedil, pravimo tudi indikativni povzetek. Povzetku, ki ga lahko preberemo namesto izvornega besedila, pravimo informativni povzetek. Informativni povzetek vsebuje dejstva, ki so navedena v izvornem besedilu, indikativni povzetek pa lahko vsebuje tudi lastnosti izvornega besedila (dolžina, slog, ipd...).

V tej disertaciji uberemo manj pogost pristop k povzemanju besedil. Naloge priprave izvršnih povzetkov se lotimo tako, da besedilo modeliramo z uporabo grafov, stavke pa izbiramo z uporabo analize z arhetipi (AA). Besedilo modeliramo na več različnih načinov: z grafom podobnosti, grafom vsebine, skupnim modelom vsebine in podobnosti ali večelementnim grafom. Stavke izbiramo z uporabo mešanice matrične dekompozicije in aproksimacijo nizkega ranga. Z drugimi besedami, izbiramo stavkov

formaliziramo kot problem analize z arhetipi. Disertacija prispeva k področju analize besedil s konkretnimi prispevki k različnim nalogam povzemanja besedil.

## *A.2 Raziskovalne teme*

Razvili smo novo metodo povzemanja besedil, ki temelji na modeliranju besedil z uporabo grafov. Raziskali smo različne naloge povzemanja besedil, natančneje: splošno, s poizvedbami, posodabljanje povzetka in primerjalno povzemanje.

*Splošno povzemanje.* Nalogi splošnega povzemanja besedil je bilo namenjenih že veliko raziskav. Naloga temelji na nekaj preprostih predpostavkah o namenu povzetka. Pri tem ne predpostavljamo nič o tipu besedil, zato vse informacije črpamo iz vhodnih besedil. Glavna predpostavka je, da morajo povzetki bralcu omogočiti, da hitro in enostavno ugotovi, o čem govorijo besedila. Zaradi te splošnosti je naloga splošnega povzemanja zelo zahtevna, kar je privedlo do bolj specifičnih nalog povzemanja, kot sta povzemanje s poizvedbami in vodeno povzemanje. V prvem delu disertacije se ukvarjamo z nalogo splošnega povzemanja skupin besedil. V ta namen smo razvili novo metodo, ki temelji na analizi arhetipov, ter jo ovrednotili na standardnih množicah podatkov, da bi preverili, kako dobro povzema besedila. Raziščemo, če lahko analizo z arhetipi uporabimo pri splošnem povzemanju, če jo lahko uporabimo za izbiro stavkov pri modelu za povzemanje, ki temelji na grafih, in, če jo lahko uporabimo pri skupnih modelih vsebine in z uporabo grafov. Prav tako raziščemo, če ima tak pristop uporabno vrednost in če je dovolj učinkovit.

*Povzemanje s poizvedbami.* Naloga takšnega povzemanja je iz besedil povzeti informacije, ki so povezane z določeno uporabnikovo poizvedbo. Na primer, če imamo uporabnikovo poizvedbo in množico relevantnih besedil, ki jih vrne spletni iskalnik, lahko skupni povzetek teh besedil olajša postopek izpolnjevanja potrebe, ki jo je uporabnik izrazil s poizvedbo. Povzemanje s poizvedbami je uporabno tudi za ustvarjanje kratkih povzetkov posameznih besedil, ki jih vrne spletni iskalnik. V tretjem poglavju predstavimo novo metodo za poizvedbeno povzemanje besedil, ki temelji na uteženi analizi za arhetipi. Raziščemo, če lahko uteženo analizo z arhetipi uporabimo za izbiro stavkov, ki je osredotočena na poizvedbo. Prav tako raziščemo, kako metoda deluje v navezi z različnimi pristopi modeliranja z uporabo grafov.

*Posodabljanje povzetka in primerjalno povzemanje.* Naloga posodabljanja povzetka je povzemanje množice novih besedil ob predpostavki, da je uporabnik že prebral in povzel neko množico besedil. Priprava posodobljenega povzetka zahteva rešitev pro-

blema zajemanja informacije, ki se spreminja skozi čas - dodajanje novih informacij v povzetek, ne da bi dodali odvečne ali že znane informacije. Primerjalno povzemanje - povzemanje razlik med dvema skupinama primerljivih besedil - je prvič predstavljeno v [3]. V četrtem poglavju predlagamo novo ogrodje MDS z uporabo utežene hierarhične analize za arhetipi. Veliko znanih nalog povzemanja besedil, vključno s splošnim povzemanjem, povzemanjem s poizvedbami, posodabljanje povzetkov in primerjalno povzemanje, lahko modeliramo kot različice predlaganega ogrodja. Raziščemo, če je predlagano ogrodje primerno in dovolj uspešno za reševanje teh nalog.

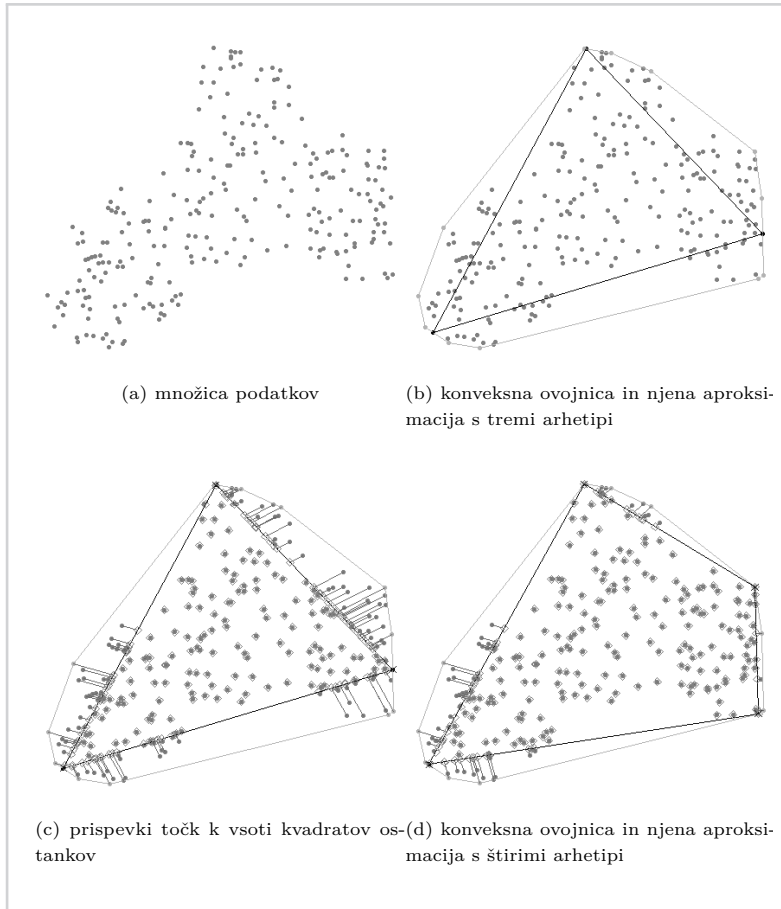
### A.3 Analiza z arhetipi

Pri analizi z arhetipi (AA), kot sta jo predstavila Cutler in Breiman [33], vsako točko iz množice podatkov ocenimo s kombinacijo točk - arhetipov, ki niso nujno točke iz množice podatkov. Arhetipi so točke iz množice podatkov ali mešanice dveh ali več točk v množici podatkov in ležijo na (ocenjeni) konveksni ovojnici množice podatkov (glejte sliko A.1).

AA odpira zanimive možnosti v podatkovnem rudarjenju, saj gre za vmesni model med aproksimacijo nizkega ranga in razvrščanjem v skupine. Koeficienti arhetipnih vektorjev se nahajajo v simpleksu, kar omogoča mehko razvrščanje v skupine, verjetnostno rangiranje ali klasifikacijo z latentnimi modeli razredov. AA je bila uporabljena na različnih področjih, kot so ekonomija [34], astrofizika [35] in razpoznavanje vzorcev [36]. Uporabnost AA za pridobivanje značilik in zmanjševanje razsežnosti prostora podatkov pri aplikacijah strojnega učenja na različnih praktičnih problemih je predstavljena v [37]. Bolj podrobno razlago numeričnih lastnosti, stabilnosti, računske zahtevnosti in implementacijo AA najdemo v [38].

#### A.3.1 Splošno povzemanje besedil z uporabo analize z arhetipi

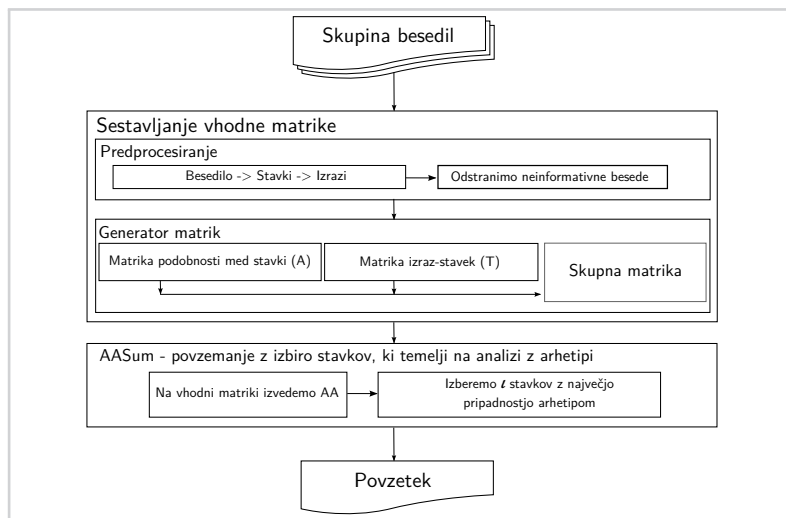
V tem razdelku predstavimo metodo za povzemanje skupin besedil AASum, pri kateri za izbiro stavkov uporabimo analizo z arhetipi. Pri modeliranju besedil z grafi segmente besedil predstavimo z vozlišči, s povezavami med njimi predstavimo informacijo o tem, kako so segmenti povezani med seboj, metrike za grafe pa opisujejo pomembnost posameznih segmentov besedila. Metoda AASum je v tem smislu nadgradnja standardnega modela z uporabo grafov, saj uporablja skupno matriko vsebine in podobnosti. Na predstavitev z vsebino in podobnostjo lahko gledamo kot na graf, v katerem je vsak stavek povezan s stavki, ki imajo porazdelitev na  $[T \times A]$  izrazih.



*Slika A.1*

Analiza z arhetipi je aproksimacija konveksne ovojnice množice podatkov. Večje število arhetipov zboljša aproksimacijo. Točke znotraj konveksne ovojnice lahko predstavimo z linearno kombinacijo arhetipov, točke izven ovojnice pa z najbližjo točko na aproksimirani obojnici. Ustrezne arhetipe poiščemo z iterativno minimizacijo ostanov točk zunaj ovojnice.





Slika A.2

Povzemanje skupin besedil z uporabo analize z arhetipi.

Metoda AASum je podrobneje razložena na sliki A.2, pri čemer z predstavlja število stavkov v povzetku. Osnovna ideja metode je preprosta: stavke z mehkim razvrščanjem razvrstimo v skupine, ki jih predstavljajo arhetipi, s čimer jih rangiramo in izberemo z najpomembnejših.

Ogrodje predlagane metode za nenadzorovano povzemanje skupin dokumentov AASum je sestavljeno iz naslednjih korakov:

1. Besedilo  $D$  razbijemo na  $n$  stavkov.
2. Predprocesiranje.
  - i Stavke razbijemo na besede.
  - ii Odstranimo besede brez informacije (angl. stop word).
3. Ustvarimo vhodno matriko  $X$ .
  - i Ustvarimo matriko podobnosti med stavki  $A$ .
  - ii Ustvarimo matriko pojavitev izrazov v posameznih stavkih  $T$ .
  - iii Vrnemo zmnožek matrik  $A$  in  $T$ .



4. Na matriki  $X$  izvedemo  $AA$ .
  - i Z  $AA$  ocenimo elemente matrične dekompozicije  $S, C$  in  $X^T C$ .
  - ii Za vsak arhetip  $i$  izračunamo njegovo pomembnost - vsoto vrednosti v ustreznem stolpcu matrike  $X^T C$ ,  $Sa_i = \sum_{j=1}^m X^T C_{j,i}$ .
  - iii Arhetipe uredimo padajoče po pomembnosti. Z drugimi besedami, stolpce matrike  $C$  uredimo po vrednostih  $Sa_i$ .
  - iv Odstranimo  $\epsilon$  najmanj pomembnih arhetipov in vrnemo rezultat.
5. Izberemo  $l$  stavkov, ki imajo največjo pripadnost najpomembnejšim arhetipom.
  - i Začnemo z najbolj pomembnim arhetipom (prvo vrstico urejene matrike  $W$ ) in stavek z najvišjo pripadnostjo temu arhetipu vključimo v povzetek. Nato izberemo drugi najbolj pomemben arhetip in postopek nadaljujemo, dokler ne izberemo zadostno število stavkov. Če pri tem obiščemo vse arhetipe, postopek nadaljujemo pri najpomembnejšem in drugemu stavku po pripadnosti temu arhetipu, ipd...
  - ii Stavkov, ki so preveč podobni kateremu izmed predhodno izbranih, ne vključimo v povzetek.

Pri tem  $\epsilon$  predstavlja število najmanj pomembnih arhetipov. V zgoraj opisanem algoritmu sta ključna četrti in peti korak. Naš namen je razvrstiti stavke v arhetipe in nato izbrati stavke z najvišjimi stopnjami pripadnosti.

Ker vsak stavek prispeva k identifikaciji vsakega izmed arhetipov, ima lahko različne vrednosti v vrsticah matrike  $C$ . Torej, stavek ima lahko visoko pripadnost k enemu arhetipu in nizko pripadnost k nekemu drugemu arhetipu. Naš cilj je izbrati najpomembnejše stavke, zato bo stavek izbran za povzetek samo, če ima visoko pripadnost k enemu izmed pomembnih arhetipov. Izstopajoči stavki bodo do četrtega koraka bolj verjetno razvrščeni v pomembnejše arhetipe. Ker stavki z višjo pripadnostjo rangirajo višje, do petega koraka izluščimo najbolj reprezentativne stavke.

Omeniti velja, da se dejstva z višjimi utežmi pojavijo v večjem številu stavkov. Torej, z analizo z arhetipi take stavke, ki si delijo skupna dejstva, razvrstimo v arhetipe z višjimi utežmi. V petem koraku izbiramo stavkov začnemo pri najpomembnejšem arhetipu in s tem zagotovimo, da avtomatski povzetek najprej pokrije najbolj utežena dejstva.

Predlagana metoda na ta način optimizira dva pomembna vidika povzemanja - pomembnost in pokritost vsebine. Pomembna funkcija teh dveh korakov je tudi zagotavljanje raznolikosti. To je deloma zagotovljeno že s pristopom z arhetipi, vendar, da uspešno odstranimo odvečne stavke in povečamo raznolikost informacije v povzedku, uporabimo požrešen algoritem (glejte korak 5.ii).

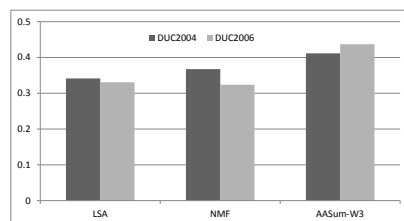
## *A.4 Poskusi*

V tem razdelku predstavimo rezultate poskusov na dveh množicah podatkov DUC, s katerimi ocenimo uspešnost predlaganih metod in njihov prispevek, v primerjavi z obstoječimi sistemi za povzemanje besedil.

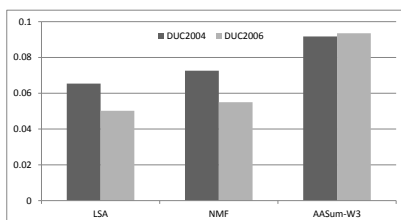
### *A.4.1 Primerjava s sorodnimi metodami.*

Predlagano metodo AASum najprej primerjamo z najbolj relevantnima metodama LSA in NMF, ki temeljita na matrični dekompoziciji [15]. Slika A.3 prikazuje, da je metoda NMF bolj uspešna od metode LSA, kar je v skladu z rezultati v [15]. Glavna prednost metode NMF je sposobnost izbire pomensko bolj primernih stavkov z uporabo semantičnih značilnk, ki jih je lažje interpretirati, in z boljšim modeliranjem zgradbe besedil. Naš pristop kaže še boljše rezultate (glejte sliko A.3), kar lahko pripišemo temu, da z uporabo analize z arhetipi bolj uspešno razvrstimo in rangiramo stavke. Slika A.3 prikazuje izboljšave metode AASum glede na metodi LSA in NMF. Vidimo, da metoda AASum daje boljše rezultate. Ker LSA in NMF temeljita na matrični faktorizaciji, lahko uspešnost metode AASum razložimo tudi s tem, da analiza z arhetipi združuje razvrščanje v skupine in matrično faktorizacijo.

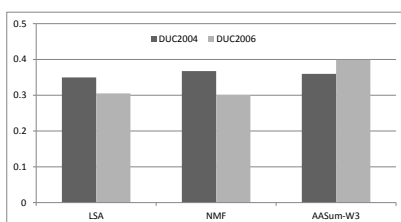
Poleg s tema metodama, smo metodo AASum primerjali tudi z nekaterimi drugimi metodami. Čeprav je na tekmovanju DUC vsako leto sodelovalo več kot 30 sistemov za povzemanje, smo se tu primerjali le z nekaterimi najboljšimi. Prednosti našega pristopa so jasno prikazane v tabelah A.1 in A.2. Rezultati predlaganega pristopa so primerljivi z najboljšimi in boljši od veliko metod v obeh letih. Kar je najbolj pomembno, naš pristop je boljši od najboljšega sistema na DUC2006 (ROUGE-1) in med najboljšimi na DUC2004.



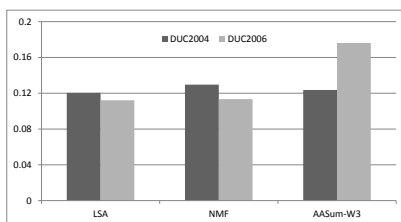
(a) ROUGE-1



(b) ROUGE-2



(c) ROUGE-L



(d) ROUGE-W

*Slika A.3*

Skupna uspešnost povzemanja na množicah podatkov DUC2004 in DUC2006.

Tabela A.1

Ocena uspešnosti metod na množici podatkov DUC2004. Opomba: "-" označuje, da avtorji niso objavili rezultatov.

Metoda	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3242	0.0641	-
Best-Human	0.3308	0.0502	0.1023
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-W <sub>3</sub>	0.4115*	0.0934	0.1376*

Tabela A.2

Ocena uspešnosti metod na množici podatkov DUC2004.

Metoda	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-W <sub>3</sub>	0.4291*	0.0944	0.1680

## A.5 Zaključek

V tej disertaciji smo opisali več prispevkov k reševanju klasičnih in novih nalog povzemanja besedil. Predlagane izboljšave izhajajo iz uporabe grafov pri modeliranju in uporabe analize z arhetipi. Večina predlaganih metod je bila razvitih izključno za povzemanje skupin besedil.

### A.5.1 Prispevki znanosti

Naše delo prispeva nove metode za povzemanje besedil (prispevki 1-3), nove metode za modeliranje vhodnega besedila z uporabo grafov (prispevka 4 in 5) in predstavlja korak naprej na področju nalog povzemanja, vključno s splošnim povzemanjem, povzemanjem s poizvedbami, posodabljanju povzetkov in primerjalnem povzemanju.

1. *Metoda za povzemanje besedil z uporabo analize z arhetipi (AASum).* Metoda AASum predstavlja pristop z uporabo analize z arhetipi pri izbiri reprezentativne in raznovrstne podmnožice stavkov iz besedila.
  - Predlagamo novo uporabo metode analize z arhetipi (AA) pri usmerjanju iskanja reprezentativnih stavkov glede na njihovo oddaljenost od pozitivno/negativno izstopajočih arhetipnih stavkov, kot jih identificira metoda AA.
  - Razvijemo učinkovit algoritem za izbiranje stavkov za povzetek, ki temelji na AA.
  - S poskusi pokažemo uspešnost predlagane metode pri nalogi splošnega povzemanja.
2. *Metoda za povzemanje besedil z uporabo utežene analize z arhetipi (wAASum).* Metoda wAASum predstavlja utežen pristop k uporabi analize z arhetipi pri izbiri reprezentativne in raznovrstne podmnožice stavkov iz besedila pri podani poizvedbi uporabnika.
  - Predlagamo novo uporabo metode utežene analize z arhetipi (wAA) pri usmerjanju iskanja reprezentativnih stavkov glede na njihovo oddaljenost od pozitivno/negativno izstopajočih stavkov, kot jih identificira metoda wAA.

- Razvijemo učinkovit algoritem za izbiranje stavkov pri povzemanju s poizvedbami, ki temelji na wAA.
  - S poskusi pokažemo uspešnost predlagane metode pri povzemanju s poizvedbami.
3. *Metoda za povzemanje besedil z uporabo utežene hierarhične analize z arhetipi (wHAASum).* Metoda wHAASum predstavlja utežen in hierarhičen pristop k uporabi analize z arhetipi pri povzemanju besedil.
- Predlagamo novo različico problema analize z arhetipi. Kolikor vemo, problem hierarhične wAA še ni bil omenjen ali raziskan.
  - Predlagamo novo uporabo metode utežene hierarhične analize z arhetipi (wHAA) pri usmerjanju iskanja reprezentativnih stavkov glede na njihovo oddaljenost od "najboljših" stavkov, kot jih identificira metoda wHAA.
  - Razvijemo ogrodje za učinkovito izvajanje vseh znanih nalog povzemanja, vključno s splošnim povzemanjem, povzemanjem s poizvedbami, posodabljanjem povzetka in primerjalnim povzemanjem.
  - S poskusi pokažemo uspešnost predlaganega ogrodja za povzemanje.
4. *Skupni model vsebine in podobnosti.*
- Skupni model vsebine in podobnosti je nov način modeliranja vhodnega besedila pri problemu povzemanja besedil.
  - Z njim na sistematičen način združimo informacije o strukturi podobnosti med izrazi in stavki.
  - Pokažemo, da metoda AASum deluje veliko bolj uspešno, če uporabimo skupni model vsebine in podobnosti.
5. *Model z uporabo večelementnega grafa.*
- Predstavimo modeliranje besedil in poizvedb z uporabo večelementnega grafa.
  - Pokažemo, da je metoda wAASum uspešna, če za model uporabimo večelementni graf.

### *A.5.2 Nadaljnje delo*

Delovanje metod za povzemanje, ki smo jih predstavili v tem delu, pri nalogi vodenega povzemanja lahko potencialno izboljšamo tako, da uporabimo označevanje semantične vloge in/ali tehnike rangiranja oz. izbiranja delov stavkov.







## BIBLIOGRAPHY

- [1] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.
- [2] Kathleen R McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc., 2002.
- [3] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Comparative document summarization via discriminative sentence selection. *TKDD*, 6(3):12, 2012.
- [4] John F Sowa. Conceptual structures: information processing in mind and machine. 1983.
- [5] Rada Mihalcea and Dragomir Radev. *Graph-based natural language processing and information retrieval*. Cambridge University Press, 2011.
- [6] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479, 2004.
- [7] Jahna Otterbacher, Gunes Erkan, and Dragomir R Radev. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing & Management*, 45(1):42–54, 2009.
- [8] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4. Barcelona, Spain, 2004.
- [9] Rada Mihalcea and Hakan Ceylan. Explorations in automatic book summarization. In *EMNLP-CoNLL*, pages 380–389, 2007.
- [10] Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation and singular value decomposition based multi-document summarization. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 713–718. IEEE, 2008.
- [11] Josef Steinberger and Karel Ježek. Text summarization and singular value decomposition. In *Advances in Information Systems*, pages 245–254. Springer, 2005.
- [12] Chang Beom Lee, Min Soo Kim, and Hyuk Ro Park. Automatic summarization based on principal component analysis. In *Progress in Artificial Intelligence*, pages 409–413. Springer, 2003.
- [13] J. Yeh. Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing & Management*, 41(1):75–95, January 2005. ISSN 03064573. doi: 10.1016/j.ipm.2004.04.003. URL <http://dx.doi.org/10.1016/j.ipm.2004.04.003>.
- [14] Yihong Gong and Xin Liu. 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*, pages 19–25. ACM, 2001.
- [15] Ju-Hong Lee, Sun Park, Chan-Min Ahn, and Daeho Kim. Automatic generic document summarization based on non-negative matrix factorization. *Information Processing & Management*, 45(1):20–34, 2009.
- [16] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. 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*, pages 307–314. ACM, 2008.
- [17] Jian-Ping Mei and Lihui Chen. Sumcr: a new subtopic-based extractive approach for text summarization. *Knowledge and information systems*, 31(3): 527–545, 2012.

- [18] Yulia Ledeneva, René García Hernández, Romyna Montiel Soto, Rafael Cruz Reyes, and Alexander Gelbukh. Em clustering algorithm for automatic text summarization. In *Advances in Artificial Intelligence*, pages 305–315. Springer, 2011.
- [19] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.
- [20] Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems*, 22(2). doi: 10.1007/s10115-009-0194-2.
- [21] Qiaozhu Mei, Jian Guo, and Dragomir Radev. 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*, pages 1009–1018. ACM, 2010.
- [22] Xiaojin Zhu, Andrew B. Goldberg, Jurgen Van, and Gael D. Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Physics Laboratory – University of Washington*, pages 97–104, 2007. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.251>.
- [23] Matthew Richardson and Pedro Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. *Advances in neural information processing systems*, 14:1441–1448, 2002.
- [24] Harendra Bhandari, Masashi Shimbo, Takahiko Ito, and Yuji Matsumoto. Generic text summarization using probabilistic latent semantic indexing. In *Proceedings of IJCNLP*, pages 133–140, 2008.
- [25] Rasim M Alguliev, Ramiz M Aliguliyev, and Makrufa S Hajirahimova. Gendocsum+ mcl: Generic document summarization based on maximum coverage and less redundancy. *Expert Systems with Applications*, 2012.
- [26] Rasim M Alguliev, Ramiz M Aliguliyev, and Nijat R Isazade. Cdds: Constraint-driven document summarization models. *Expert Systems with Applications: An International Journal*, 40(2):458–465, 2013.
- [27] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Chingiz A. Mehdiyev. An optimization approach to automatic generic document summarization. *Computational Intelligence*, 29(1):129–155, 2013.
- [28] Wenjie Li, Baoli Li, and Mingli Wu. Query focus guided sentence selection strategy for duc 2006. In *Proceedings of Document Understanding Conferences*, 2006.
- [29] Sun Park, Ju-Hong Lee, Chan-Min Ahn, Jun Sik Hong, and Seok-Ju Chun. Query based summarization using non-negative matrix factorization. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 84–89. Springer, 2006.
- [30] Ziheng Lin, Tat-Seng Chua, Min-Yen Kan, Wee Sun Lee, Long Qiu, and Shiren Ye. Nus at duc 2007: Using evolutionary models of text. In *Proceedings of Document Understanding Conference (DUC)*, 2007.
- [31] Li Wenjie, Wei Furu, Lu Qin, and He Yanxiang. Pnr 2: ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 489–496. Association for Computational Linguistics, 2008.
- [32] Ruifang He, Bing Qin, and Ting Liu. A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering. *Expert Systems with Applications*, 39(3):2375–2384, 2012.
- [33] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.
- [34] Giovanni C Porzio, Giancarlo Ragozini, and Domenico Vistocco. On the use of archetypes as benchmarks. *Applied Stochastic Models in Business and Industry*, 24(5):419–437, 2008.
- [35] Ben HP Chan, Daniel A Mitchell, and Lawrence E Cram. Archetypal analysis of galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 338(3):790–795, 2003.
- [36] Christian Bauchhage and Christian Thureau. Making archetypal analysis practical. In *Pattern Recognition*, pages 272–281. Springer, 2009.
- [37] Morten Morup and Lars Kai Hansen. Archetypal analysis for machine learning and data mining. *Neurocomputing*, 80:54–63, 2012.
- [38] Manuel Eugster and Friedrich Leisch. From spiderman to hero-archetypal analysis in r. 2009.
- [39] Manuel JA Eugster and Friedrich Leisch. Weighted and robust archetypal analysis. *Computational Statistics & Data Analysis*, 55(3):1215–1225, 2011.
- [40] Ercan Canhasi and Igor Kononenko. Multi-document summarization via archetypal analysis of the content-graph joint model. *Knowledge and Information Systems (in press)*. doi = 10.1007/s10115-013-0689-8.
- [41] Inderjeet Mani. *Automatic summarization*, volume 3. John Benjamins Publishing Company, 2001.
- [42] Mohamed Abdel Fattah and Fuji Ren. Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech & Language*, 23(1):126–144, 2009.

- [43] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.
- [44] Chin-Yew Lin and Eduard Hovy. 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-Volume 1*, pages 71–78. Association for Computational Linguistics, 2003.
- [45] Ercan Canhasi and Igor Kononenko. Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems with Applications (in press)*. doi = 10.1016/j.eswa.2013.07.079.
- [46] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. 1999.
- [47] Deepak Verma, Karl Pflieger, and David Tax. Recursive attribute factoring. In *Advances in Neural Information Processing Systems*, pages 297–304, 2006.
- [48] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
- [49] Hoa Trang Dang and Karolina Owczarzak. Overview of the tac 2008 update summarization task. In *Proceedings of text analysis conference*, pages 1–16, 2008.
- [50] Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Comparative document summarization via discriminative sentence selection. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *CIKM*, pages 1963–1966. ACM, 2009. ISBN 978-1-60558-512-3.
- [51] Duncan McCallum and David Avis. A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters*, 9(5):201–206, 1979.
- [52] Rex A Dwyer. On the convex hull of random points in a polytope. *Journal of applied probability*, pages 688–699, 1988.
- [53] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10): 2756–2779, 2007.
- [54] Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. Automated summarization evaluation with basic elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006)*, pages 604–611. Citeseer, 2006.
- [55] Stephen Tratz and Eduard Hovy. Summarization evaluation using transformed basic elements. In *Proceedings of the 1st Text Analysis Conference (TAC)*. Citeseer, 2008.
- [56] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics, 2005.
- [57] Pierre-Etienne Genest, Guy Lapalme, and Mehdi Youfi-Monod. Hextac: the creation of a manual extractive run. In *Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology*, 2009.
- [58] Ercan Canhasi and Igor Kononenko. Weighted hierarchical archetypal analysis based generic multi-document summarization framework. *Technical report, 2013 (submitted for publication)*.