A Genetic Clustering Algorithm for Automatic Text Summarization

Sebastian Suaréz Benjumea

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Sistemas e Industrial

Grupo de Investigación MIDAS

Bogotá, Colombia

2015

Genetic Clustering Algorithm for Extractive Text Summarization

Sebastian Suaréz Benjumea

A thesis submitted in partial fulfillment of the requirements for the degree of:

Master in Systems and Computer Engineering

Advisor:
Elizabeth León Guzmán , Ph.D.

Research Area:
Text Mining, NLP

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Sistemas e Industrial
Grupo de Investigación MIDAS
Bogotá, Colombia
2015

Dedicación

A Gloria Ines y Hector Jose mis padres que siempre me dieron la libertad de escoger, con su comprensión y apoyo me permitieron dedicar mi tiempo a la ciencia.

A mis padres

Gloria Ines Benjumea
Hector Jose Suarez

Acknowledgment

I would like to thank my advisor professor Elizabeth León Guzmán and the MIDAS research group for their support in this thesis.

I would like to thank National University of Colombia for their support.

Abstract

Automatic text summarization has become a relevant topic due to the information overload. This automatization aims to help humans and machines to deal with the vast amount of text data (structured and un-structured) offered on the web and deep web. In this research a novel approach for automatic extractive text summarization called SENCLUS is presented. Using a genetic clustering algorithm, SENCLUS clusters the sentences as close representation of the text topics using a fitness function based on redundancy and coverage, and applies a scoring function to select the most relevant sentences of each topic to be part of the extractive summary. The approach was validated using the DUC2002 data set and ROUGE summary quality measures. The results shows that the approach is representative against the state of the art methods for extractive automatic text summarization.

Resumen

La generación automática de resúmenes se ha posicionado como un tema de gran importancia debido a la sobrecarga informativa. El objetivo de esta tecnología es el ayudar humanos y maquinas a lidiar con el gran volumen de información en forma de texto (estructurada y no estructurada) que se encuentra en la red y en la red profunda. Esta investigación presenta un nuevo algoritmo para la generación automática de resúmenes extractivos llamado SENCLUS. Este algoritmo es capaz de detectar los temas presentes en un texto usando una técnica de agrupación genética para formar grupos de oraciones. Estos grupos de oraciones son una representación aproximada de los temas del texto y estos son formados usando una función aptitud basada en cobertura y redundancia. Una vez los grupos de oraciones son encontrados, se aplica una función puntuación para seleccionar las oraciones mas relevantes de cada tema hasta que las restricciones de longitud del resumen lo permitan. SENCLUS fue validado en una serie de experimentos en los cuales se usò el conjunto de datos DUC2002 para la generación de resúmenes de un solo documento y se usò la medida ROUGE para medir de forma automática la calidad de cada resumen. Los resultados mostraron que el enfoque propuesto es representativo al ser comparado con los algoritmos presentes en el estado del arte para la generación de resúmenes extractivos.

Keywords: text mining, genetic algorithm, clustering algorithm, automatic text summarization, single document automatic text summarization

Contents

List of Figures

List of Tables

1 Introduction

Nowadays, the volume of text data is a lot bigger than 10 years ago. With the establishment of the web 2.0, Twitter, Facebook, online forums, social networks, blogs, self-newspaper (run by individuals and not big media companies) and others, the task of extracting value of such data maze becomes more important. This immense amount of digital data presents an obstacle for people reading it, so better tools that help them to cope with the information overload are expected.

The objective of the text summarization is, "*obtain a reductive transformation of the base text to summarize via condensation, applying generalization and/or particularization of what it is important in the base text*" [23]. However this functional definition is incomplete because it does not consider the particular interests of the user, which affect the usefulness of the summary. A better definition could be given by combining the previous definition with the one given in [56]: "*The text summarization aims to produce a brief but accurate representation of the most important information present in the base text to satisfy a set of user/users information requirements*". Additionally, this definition has to deal with the fact that humans are not sure about what information should be in the summaries [47], as they are not able to foresee readerships interests and expectations. Then, automatizing text summarization as well as the ways to validate it automatically become a difficult problem that requires new approaches to be solved.

The Automatic Text Summarization (ATS) is simply an automatic implementation of the text summarization applied to large volumes of documents (source text) to help humans and machines to cope with the vast amount of (structured and un-structured) data present on the web and deep-web[1].

Depending on the summary form it could be an extract or an abstract. The extract summary is composed by exact words or phrases which are present in the source text. The abstract summary is composed by words, phrases or expressions that are not necessarily present in the source text; this type of summary is strongly related with the text understanding. Different techniques have been used to solve the extractive ATS problem, including statistical based, graph based, classification based, and bio-inspired [34, 39, 56, 62] techniques. However, some of them take into account the possibility of exploring clustering techniques [68, 48, 59].

[1] The Deep Web (also called the Deep-net, the Invisible Web, the Undernet or the hidden Web) is World Wide Web content that is not part of the Surface Web, which is indexed by standard search engines.

Clustering techniques can be used to solve the extractive ATS problem because it can be modeled as a multimodal optimization problem in which the algorithm aims to find the best sentences clusters. This sentences clusters are made of redundant sentences so an extractive summary should have the best sentences of each cluster taking into account the cluster relevance. Several multimodal optimization problems has been solved using an Evolutionary Algorithms (EA) or niching strategies showing good results [61]. But, some EA have the tendency to lose diversity within their population of feasible solutions and to converge into a single solution. The niching strategies address this issue by maintaining the diversity of certain properties within the population and this way they allow parallel convergence into multiple good solutions in multimodal domains.

An extractive summary could be generated from a single document or from multiple documents. The multimodal multidocument optimization problem is harder than the single document problem because on the first one the sentences diversity is bigger which causes an increase on the number of solutions. Also, the size of multimodal domain for the multidocument summarization problem is higher. On the other hand, the mutimodal domain and sentences diversity is smaller for the single document problem and more important the ideas and conclusions obtained from studying it could help to solve the multidocument summarization problem. Indeed, both problems are very similar and the most important difference between them is the composition and size of the search space. Then, make an study on the single document summarization problem is the best choice because it is less complex and the studies over it could contribute to solve the multidocument problem.

As it was metioned, a multimodal problem can be solved using niching techniques. Evolutionary Clustering with Self Adaptative Genetic Operators (ECSAGO) [29] is a self adaptative clustering algorithm that uses a niching technique and is robust to noise. This algorithm has been used for clustering text showing good results [28, 27] and this research propose an ECSAGO adaptation specifically designed to solve the problem of generate extractive summaries automatically by using a different fitness function based on redundancy and coverage. This new algorithm called SENCLUS is a genetic clustering algorithm for single document extractive automatic text summarization. The algorithm uses a genetic clustering technique with a fitness function based on coverage and redundancy to automatically detect the text topics generating good extractive text summaries which cover the most important text topics with little algorithm configuration parameters. SENCLUS is capable of generate extractive summaries which are statistically representative against the state of the art algorithms for single document summarization.

1.1 Goals

The purpose of this work is to develop a genetic clustering algorithm for generating single document extractive summaries, extending the Evolutionary Clustering Algorithm with

Parameter Adaptation (ECSAGO). To aims this goal, it is proposed specifically:

1. **Making literature review:** This work presents review of the different techniques used for automatic text summarization. To carry out this, a literature review is performed. This review is focused on techniques and algorithms for extractive summarization. A state of the art is developed and presented.

2. **Designing and Implementing a genetic algorithm for the sentences selection:** This work presents a genetic clustering algorithm for sentences clustering. This algorithm is based on the ECSAGO algorithm. The algorithm includes mechanisms for topics detection based on sentences clustering using a fitness function based on coverage and redundancy.

3. **Compare and asses the algorithm:** This work presents a summary of the conducted experiments for generate single document extractive summaries using the DUC2002 data set. The generated summaries were validated using the ROUGE measure. Finally, the reported results were compared against the state of the art algorithms that reported results for the same data set.

Methodology

The algorithm was developed using and iterative methodology in which each iteration consist of design, algorithm codification, experiments, experiments validations and conclusions. After each iteration the detected problems will be tackled on the next iteration.

1.2 Contributions

1. A state of the art for Automatic Text Summarization.

2. A new algorithm for single document Automatic Extractive Text Summarization based sentences clustering.

    a) A topics detection model based on sentences clustering.

    b) An adaptation of ECSAGO for Automatic Text Summarization called SENCLUS.

    c) A fitness function based on redudancy and coverage used by SENCLUS.

    d) A cluster radius used by SENCLUS to allow niching and delimit the topics.

3. Article: “Genetic Clustering Algorithm for Extractive Text Summarization”, submitted and accepted in IEEE SSCI 2015.

1.3 Outline

The document is organized as follows: chapter 2 presents the Automatic Text Summarization problem, the approaches commonly used for it, vector space model and techniques used for validating summaries automatically. This chapter also describes the ECSAGO algorithm and other clustering algorithms for solving the summarization problem. In chapter 3, the proposed SENCLUS algorithm for single document text summarization is discussed, describing the details of the approach; in chapter 4 an analysis of the experiments results using a single document summarization data set is conducted, showing how SENCLUS results contrast with the single document summarization state of the art algorithms and other clustering algorithms. Finally in chapter 5, the conclusions and future research are presented.

2 Background

This chapter presents an Automatic Text Summarization state of the art. The section 2.1 introduces the different types of summaries providing a definition for each one. Section 2.2 offers details of the text representation used in this research. A review of techniques used for Automatic Text Summarization is presented in section 2.4 and the techniques used for evaluate automatic summaries are presented in section 2.3. Finally, section 2.5 introduces the genetic clustering ECSAGO on which SENCLUS is based and section 2.6 describes some popular clustering algorithms used for automatic text summarization.

2.1 Types of Automatic Summaries

The objective of the text summarization is, "*obtain a reductive transformation of the base text to summarize via condensation, applying generalization and/or particularization of what it is important in the base text*" [23]. In general, summarization techniques are classified over three main aspects: input, purpose and output [23, 21]. However, there are other ways for summaries classifications, for example [21] gives special attention to the coherence and subjectivity factors of the final summary; and [35] classify the summarization systems based on the approach adopted (surface, entity and discourse level). The most important summary components used for classify a summary are shown in Figure (2.1) and are explained below.

Depending on the way how the summary is composed it could be:

- **Extract:** The extract summary is composed by exact words or phrases which are present in the source text.
- **Abstract:** The abstract summary is composed by words, phrases or expressions that are not present in the source text necessarily. This type of summary is strongly related with automatic text understanding and automatic text generation .

Depending on the level of processing it could be:

- **Surface-level approaches:** These processing approaches perform a superficial analysis to produce the summaries. For example, it uses only words counts, position and other basics text statistics to obtain the summaries.

- **Deeper-level approaches:** These processing approaches apply more complex analysis over the text. For example, they apply semantic analysis and natural language generation to produce more complex summaries that could be extracts or abstracts.

Depending on the summary purpose :

- **Indicative summaries:** The indicative summaries are those summaries which have length 5-10% of the source text and give a short version of the main topics of the text. These summaries helps the user to decide if it worths to read the whole text.
- **Informative summaries:** The informative summaries give a summary having a length of 20-30% of the source text. These summaries only keep the important information of the base text.
- **Critical or evaluative summaries:** These type of summaries are the more complex and they try to retrieve the points of view or opinions of the authors present in the base text.

Depending on the audience:

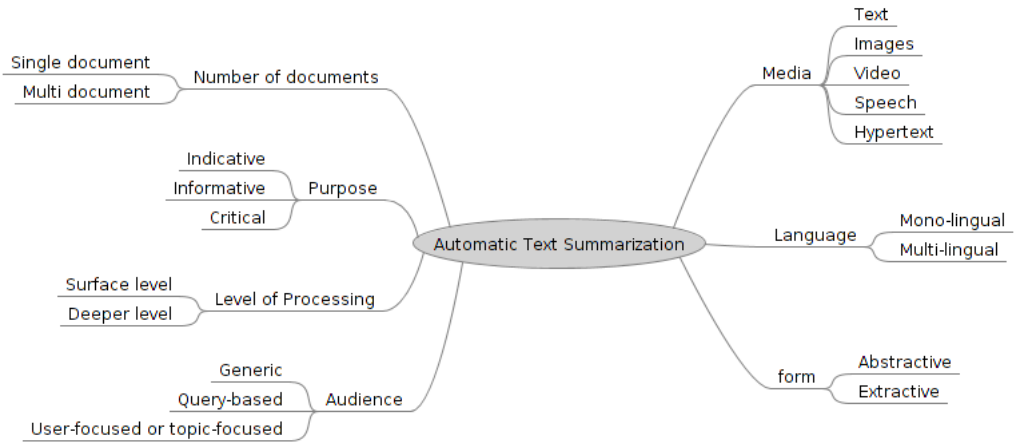
- **Generic summaries:** The generic summaries produce a summary where all the topics are equally important.
- **Query-based summaries:** These summaries aim to focus the summary on the given query (topic).
- **User focused or topic focused summaries:** This type of summaries intends to generate a summary which could inform a group of people with interest in certain topics.

Finally depending on the number of document it could be: single-document or multi-document. And depending on the languages involved in the source could be: mono-lingual or multi-lingual.

2.2 Text Documents Representation

The popularity of using vector spaces for representing text is that it provide a natural mechanism for work with concepts from geometry like distance and similarity [11]. For example, there are many aspects of semantics, particularly lexical semantics, which require a notion of distance [11]. In the vector space, the meanings of words will be represented using vectors as part of a high-dimensional "semantic space". The fine-grained structure of this space is provided by considering the contexts in which words occur in large corpora of text. Then, words could be easily compared in the vector space using any of the standard

Figure **2-1**: Summary components used to classify a summary technique



similarity or distance measures available from linear algebra, for example the cosine. Others models based on and extending the vector space model include:

- Generalized vector space model
- Latent semantic analysis
- Term Discrimination Rocchio
- Classification Random Indexing

This section introduces details of the vector space representation along with *term-frequency* and *inverse-document-frequency.*

2.2.1 Vector Space Representation

Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, such as, for example, index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings. It represents each document as a vector with one real-valued component, usually a $tf - idf$ weight, for each term. The representation of a set of documents as vectors in a common vector space is known as the vector space model and is fundamental to a host of information retrieval operations ranging from scoring documents on a query, document classification and document clustering.

Vector-space models rely on the premise that the meaning of a document can be derived from the document's constituent terms. They represent documents as vectors of terms $d = \{t_1, t_2, \ldots, t_n\}$ where $t_i \, (1 \leq i \leq m)$ is a non-negative value denoting the single or multiple

occurrences of term i in document d. Thus, each unique term in the document collection corresponds to a dimension in the space. Both the document vectors and the query vector provide the locations of the objects in the term-document space. By computing the distance between the query and other objects in the space, objects with similar semantic content to the query presumably will be retrieved. Vector space models are more flexible than inverted indices since each term can be individually weighted, allowing that term to become more or less important within a document or the entire document collection as a whole. Also, by applying different similarity measures to compare queries to terms and documents, properties of the document collection can be emphasized or deemphasized. For example, the dot product (or, inner product) similarity measure finds the Euclidean distance between the query and a term or document in the space. The cosine similarity measure, on the other hand, by computing the angle between the query and a term or document rather than the distance, deemphasizes the lengths of the vectors[30].

2.2.2 Term Frequency and Inverse Document Frequency

An intuitive way of representing text in the vector space is use the *term frequency* (tf) to model each term. But tf can not be used to discern among common used terms and relevant terms. In cases where only tf is used, terms that are used indifferently are more relevant than terms which distinguish a document from the others. Then, a weighted tf value known as $tf - idf$ was introduced.$tf - idf$ stands for term frequency-inverse document frequency, and the $tf - idf$ weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the $tf - idf$ weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. Also, $tf - idf$ can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

Typically, the $tf - idf$ weight is composed by two terms: the first computes the normalized *term frequency* (tf) which is the number of times a word appears in a document divided by the total number of words in that document; the second term is the Inverse Document Frequency (idf) which is computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears [30].

2.3 Automatic Summaries Evaluation Measures

The methods used for evaluate the summaries quality can be broadly classified into two (2) categories: intrinsic and extrinsic. The most common approach to measure a summary

in a intrinsic way is evaluate the summary informativeness by comparing the content of a generated summary against a human-based model summary. In other words check how similar is the content of the reference summary (human-based) against the generated one. Based on this idea, there has been develop several methods. Due to the subjectivity of the summaries, is not possible to make a fair comparison between two summaries because the task of define a gold-standard is hard. It means that is not clear yet how to define a global standard summary with which the comparison is going to be done in a fair way. Also as it has been mentioned in the Chapter 1, the conception of what is a good summary varies depending on the user needs and profile. These issues add complexity to the problem of measure the quality of a summary. The remaining of this section is dedicated to talk about informativeness evaluation techniques and another dedicated to the evaluation problem are going to be presented here.

2.3.1 Informativeness evaluation

In order to asses the informativeness of a summary a well known information retrieval (IR) measures like recall, precision and F-measure has been used. The recall evaluates the number of sentences present in the generated summary, while the precision checks how many sentences are present in the both summaries, model and generated one. The F-measure is simply a mixture of the recall and precision.

In general all the metrics tries to check in the most fair way two summaries. For example Radev and Tam [50] proposed the Relative Utility, where the quality of a summary is measured using different ranks from experts for a given sentence, and comparing them against the generated ones. Teufel and Halteren [58] use factoids (atomic information units which represents the meaning of a sentence) to find the gold standard overlapping summaries of different experts for the same base text. The same idea of information overlapping is used in the Pyramid method proposed by Passoneau [49]. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) proposed by Lin [67] uses the ideas of the recall, precision and F-measure but applied at n-gram level.

A more complex approaches are used, for example QARLA. This is a evaluation framework [6] which mixes a total of 59 similarity measures to return a similarity factor between two summaries. In [20] the authors use the Basic Elements (BE), which are a words triplets obtained from the sentence which helps to give more flexibility to the matching process. Another approach called ParaEval [69] was designed to facilitate the detection of paraphrase matching.

The metric 'text grammars' takes into consideration surface and deep structures in order to describe a valid text structure in a formal way. One of the newest metric is AutomSummENG [19] which has proved a high correlation with humans judgments. In [46] the authors propose DEPEVAL which is a dependency-based metric. The idea is very similar to the

Table **2-1**: Types of informativeness evaluation methods

Approach	Automatic	Semi-automatic
Relative utility		OK
Factoid score		OK
Pyramid method		OK
ROUGE	OK	
QARLA	OK	
BE	OK	
Text grammars		OK
ParaEval	OK	
AutoSummENG	OK	
DEPEVAL	OK	
GEMS	OK	
HowNet	OK	

Source: [34]

Basic Elements (BE) approach, an uses triplets to measure the similarity between two summaries. The GEMS metric (Generative Modeling for Evaluation of Summaries) proposed by Katragadda [24] was created specifically for languages different from English. This metric uses the HowNet resource to calculate similarity. And finally the HowNet uses the WordNet databases to measures the summary quality[34].

Table (**2-1**) present a summary.

2.3.2 Rouge

ROUGE, or Recall-Oriented Understudy for Gisting Evaluation [67], is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.

The following five evaluation metrics are available.

- ROUGE-N: N-gram based co-occurrence statistics.
- ROUGE-L: 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: Weighted LCS-based statistics that favors consecutive LCSes .
- ROUGE-S: Skip-bigram based co-occurrence statistics. Skip-bigram is any pair of words in their sentence order.

- ROUGE-SU: Skip-bigram plus unigram-based co-occurrence statistics.

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as showed in equation (2-1).

$$ROUGE - N = \frac{S\{ReferenceSummaries\}\, gram_n S\, Count_{match}\,(gram_n)}{S\{ReferenceSummaries\}\, gram_n S\, Count\,(gram_n)} \tag{2-1}$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. It is clear that ROUGE-N is a recall-related measure because the denominator of the equation is the total sum of the number of n-grams occurring at the reference summary side. Note that the number of n-grams in the denominator of the ROUGE-N formula increases as we add more references. This is intuitive and reasonable because there might exist multiple good summaries.

Every time we add a reference into the pool, we expand the space of alternative summaries. By controlling what types of references we add to the reference pool, we can design evaluations that focus on different aspects of summarization. Also note that the numerator sums over all reference summaries. This effectively gives more weight to matching n-grams occurring in multiple references. Therefore a candidate summary that contains words shared by more references is favored by the ROUGE-N measure. This is again very intuitive and reasonable because we normally prefer a candidate summary that is more similar to consensus among reference summaries [67].

2.4 Techniques used for Automatic Extractive Summarization

To solve the problem of automatic summarization there have been proposed different types of solutions. In this section a short summary of bio-inspired and no bio-inspired approaches used for the automatic summarization is presented.

2.4.1 No Bio-inspired approaches

In [36] a statistical approach which uses *tf-idf* was proposed and in [33] a similar approach was presented with the difference that it removes the stops words to use the *tf* using the length as the *tf* weight. Although this approaches could be considered not sufficiently good,[45, 44] showed that those techniques despite of being simple and do not require a deep level of knowledge analysis,they are appropriate for building good summaries. Also, the mutual information which can measures the dependency or common information between

two words, information gain (metric for deciding the relevance of an attribute) and residual inverse document frequency (which is variant of the invert document frequency, computes the term document frequency according to a Poisson distribution) were used. In [43], a clusters based approach was proposed, creating groups of documents based on a similarity measure. Then using information gain selects the most important sentence of each cluster[34].

In classification based approaches the problem is reduced to assign a class to a particle (word, sentence,..): {important or not important}; this class or label is used to decide if the particle belongs to the summary or not. So in general the automatic summarization is modeled as a 2-class problem. The proposed classification techniques were: binary classifiers [26], Hidden Markov Models [13, 52], Bayesian methods [7], Neural Networks [57, 10], Support Vector Regression [51], Least Angle Regression [31], Non-Negative Matrix Factorization[48, 59] and Support Vector Machines [17, 65][34].

In the case of graph-based ranking algorithms, they have been shown to be effective solving the Automatic Text Summarization problem. The main idea is that the nodes of the graph represent text elements (words, sentences, etc). Based on the text represented as a graph, the idea is that the topology of the graph will reveal interesting patterns and features about text elements, for example the connectivity of the different elements. LexRank is used in [15, 40, 64], and an analysis over the graph is carried out to find similarities between text elements. In [18] characters and word n-grams graphs are used to extract relevant information from a set of documents, whereas in [42] graphs are built using concepts identified with WordNet [42] and its relationships, which are then used to build a graph representation for each sentence in a document[34].

2.4.2 Bio-inspired approaches

Genetic strategies has been used to solve the summarization problem. Works presented in [14, 25, 32] use Genetic approaches defining a set features $f = \{f_1, \ldots, f_n\}$ to extract the best sentences of a document optimizing the features weights w_i .

The work [14] uses eight sentence features: sentence length, similarity to the title, occurrence of non-essential information, sentence-to-centroid cohesion and others. The genetic algorithm is designed to find the best weight w_i for each feature f_i that maximizes the fitness function $f(x)$, which was defined as the average classification precision. The only differences with [32] are the use of 31 features and the support for multilingual problems. A similar approach is applied in [25] using a genetic algorithm to optimize a function with weight w_i for six features. In this work, the GA (Genetic algorithm) is used to optimize the weights while the GP (Genetic Programming) is used to optimize the set of fuzzy rules which leads to decide if a sentence should or should not belong to the summary.

In [3] the summarization problem is modeled as a *p-median* problem. The authors used a fitness function that balance the relevance, content coverage and diversity in the summary in

order to find the best combination of sentences. The optimization method used in the genetic algorithm is Differential Evolution (DE) algorithm, which is a population based stochastic search technique.

In [55] the extractive summarization problem is solved using a Fuzzy Evolutionary Optimization Modeling (FEOM) which is applied to solve the sentence clustering problem, where each cluster center is sentence of the summary.

The MCMR function is used in[1, 2, 4]. MCMR is based on the idea that a summary sentence should have a high text coverage and low redundancy against the others summary sentences so the summary sentences are the ones that maximize this function. The approach described in [2] uses PSO, showing very good results that are supported also by the results obtained in [1, 4] where DE (Differential Evolution) is used instead.

In [53] a summarization method based on harmonic search is used to extract the most relevant sentences of the source text. The authors take into account three (3) factors in the objective function: (i) Topic Relation Factor: Measures the similarity between the sentences and the text title. (ii) Cohesion Factor: Similarity between the summary sentences. (iii) Legibility Factor: Similarity of one summary sentence with the next. The used harmonic vector is of length n (total number of sentences in the document), and a binary model where 1 means that the sentence belongs to the summary and 0 otherwise.

In [16] a Genetic Algorithm is used to find the optimal values for a weight w_i for each feature f_i, where $i = 1, \ldots, 10$ using a training data set. After the training stage, the test stage is run. In this stage with a linear combination of $w_i f_i$, a new instance (sentence) is assigned a real value. The top n sentences are select to conform the final summary. In the GA a chromosome is represented as as the combination of all w_i, and a total of 100 generations selecting the 10 best individuals for the crossover process is performed to obtain the optimal individual.

In [8] an automatic summarization model which integrates fuzzy logic and swarm intelligence is proposed. The swarm model is used to calculate the values or weights w_i for the features f_i, where $i = 1, \ldots, 5$. Then the weights are used as inputs for the fuzzy inference system in order to assign a final value to the sentences, which is used to rank the sentences and select the top n sentences.

Finally, a recent memetic algorithm called MA-MultiSumm[38] has shown great results compared with the state of the art algorithms using a evolutionary algorithm to select the best sentences applying a binary optimization.

2.5 ECSAGO

Evolutionary Clustering with Self Adaptive Genetic Operators (ECSAGO) [29] is self adaptive genetic clustering algorithm that uses a niching technique. The niching technique allows

to different types of life (samples) form clusters using the domain space context for define their niches. This algorithm is able to adapt the genetic operators rates automatically at the same time it is evolving the clusters prototypes using the HAEA. Each individual represents a candidate cluster (center and scale) and while the center of the cluster is evolved using a Evolutionay Algorithm, it scale (radius) is updated using an iterative hill-climbing procedure. To preserve the detected niches, a restriction mating is imposed in which only individuals that belong to the same niche can produce offspring.

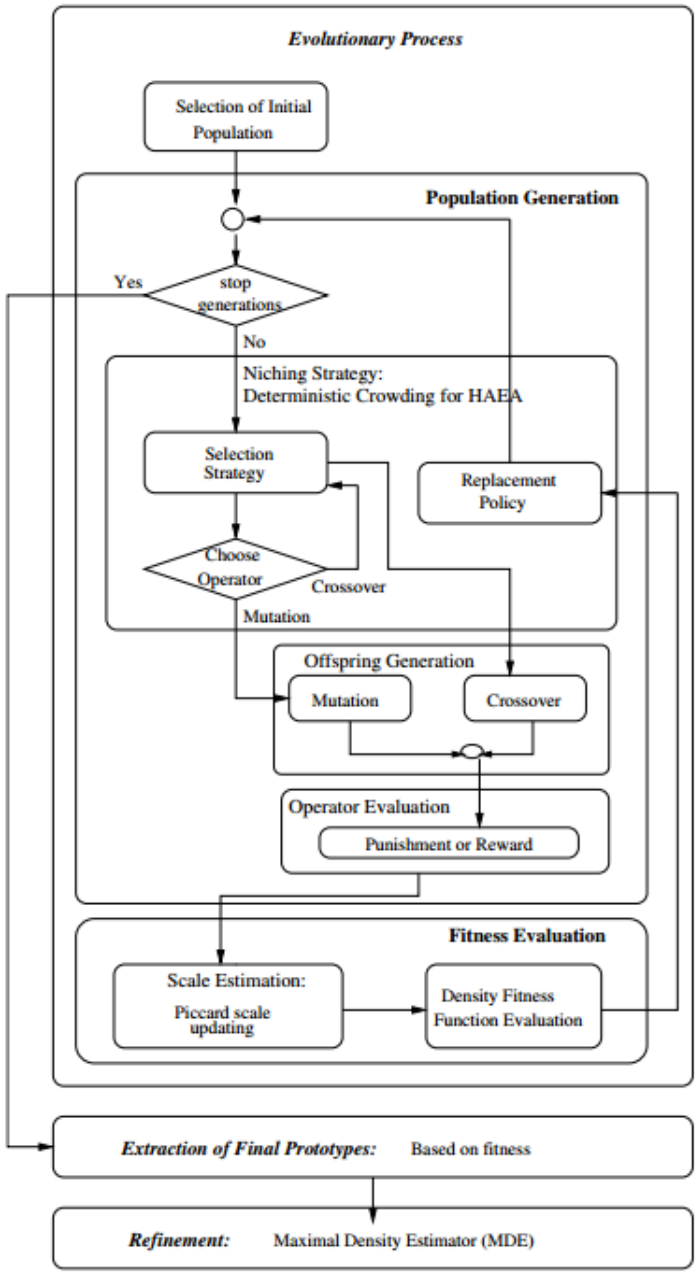
One disadvantage of the Genetic Algorithms is the genetic operator tuning which could be a time consuming task. This task consists in selecting the right group of genetic operator and the correct probability value to decide when to apply each one. To deal with the genetic operators tuning parameters of the Genetic Algorithm (GA), ECSAGO uses Hybrid Adaptive Evolutionary Algorithm (HAEA) which is a parameter adaptation technique of Evolutionary Algorithms. In HAEA, each individual is evolved independently of the other individuals in the population. In each generation, one genetic operator (crossover, mutation, etc.) is selected for each individual according to operator rates that are encoded into the individual. The fitness value f_i of i^{th} cluster candidate c_i showed in equation (2-2), is the density of the hypothetical cluster in which σ^2 is the cluster scale or size of cluster and N is the number of data points. Each cluster scale is updated after each generation of the GA using equation (2-3), where $\omega_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)$ is a robust cluster fit weight that use the distance from a data point x_i to a cluster center c_i and the σ_i^2 value of the previous generation.

$$f_i = \frac{\sum_{j=1}^{N} \omega_{ij}}{\sigma_i^2} \tag{2-2}$$

$$\sigma_i^2 = \frac{\sum_{j=1}^{N} \omega_{ij} d_{ij}^2}{\sum_{j=1}^{N} \omega_{ij}} \tag{2-3}$$

These attributes turn ECSAGO capable for detect the number of clusters of different sizes allowing it to be robust to noise and outliers, and capable to automatically adapt the genetic operators avoiding the try and error process for fixing these parameters. The ECSAGO evolutionary process is showed in Figure **2-2**. This algorithm has been used for clustering text showing good result [28, 27].

Figure **2-2**: ECSAGO [29]



2.6 Clustering Algorithms for Automatic Extractive Text Summarization

2.6.1 K-means

Clustering is the process of partitioning a group of data points into a small number of clusters. For instance, the items in a supermarket are clustered in categories (butter, cheese and milk are grouped in dairy products). Of course this is a qualitative kind of partitioning. A quantitative approach would be to measure certain features of the products, say percentage of milk and others, and products with high percentage of milk would be grouped together. In general, we have n data points $x_i, i = 1...n$ that have to be partitioned in k clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to find the positions $\mu_i, i = 1...k$ of the clusters that minimize the distance from the data points to the cluster. K-means clustering solves equation (2-4) where c_i is the set of points that belong to cluster i. The K-means clustering uses the square of the Euclidean distance $d(x, \mu_i) = \|x - \mu_i\|_2^2$. This problem is not trivial (in fact it is NP-hard), so the K-means algorithm only hopes to find the global minimum, possibly getting stuck in a different solution [66].Finally, the K-means algorithm has been used as a sentences clustering algorithm for extractive ATS reporting interesting results [68]. This K-means implementation for ATS clusters the sentences using a semantic distance among the sentences in the cluster and then calculates the accumulative sentences similarity of each cluster. The extractive summary is obtained by selecting the topic sentences using a defined set of extraction rules.

$$\arg\min_{c} \sum_{i=1}^{K} \sum_{X \in c_i} d\left(X, \mu_i\right) = \arg\min_{c} \sum_{i=1}^{K} \sum_{X \in c_i} \|X - \mu_i\|_2^2 \tag{2-4}$$

2.6.2 GK-means

K-means algorithm is the most popularly used algorithm to find a partition that minimizes total within cluster variation measure . A major problem with the K-means algorithm is that it is sensitive to the selection of initial partition and may converge to a local minimum of variation if the initial partition is not properly chosen. Since stochastic optimization approaches are good at avoiding convergence to a local optima, these approaches could be used to find a globally optimal solution. For the purpose of finding the global minimal a Genetic Algorithm (GA) is used. The genetic operators that are used in GKA are the selection, the distance based mutation and the K-means operator. With a data set of n samples, each individual of length n represents a solution in which each allele in a chromosome could take values from $1, 2, \ldots..., k$. In each iteration apply the selection, mutation and K-means operator which is a one step k-means algorithm used to reduce the oscillation behavior

of the algorithm. With this, the GK-means tries to find the global optimal avoiding the local optimal convergence [54].

2.6.3 NMF

Non-negative matrix factorization (NMF) is a linear, non-negative approximate data representation. Given a non-negative data matrix V, NMF finds an approximate factorization $V \approx WH$ into non-negative factors W and H. Let us assume that our data consists of t measurements of n non-negative scalar variables. Denoting the (n-dimensional) measurement vectors $v^t(t = 1, ..., T)$, a linear approximation of the data is given by equation (2-5), where W is an $n \times m$ matrix containing the basis vectors w_i as its columns.

$$v^t \approx \sum_{i=1}^{m} w_i h_i^t = W h^t \tag{2-5}$$

Note that each measurement vector is written in terms of the same basis vectors. The m basis vectors w_i can be thought of as the *'building blocks'* of the data, and the (m-dimensional) coefficient vector h^t describes how strongly each building block is present in the measurement vector v^t . Arranging the measurement vectors v^t into the columns of an $n \times t$ matrix V, we can now write $V \approx WH$, where each column of H contains the coefficient vector h^t corresponding to the measurement vector v^t . Written in this form, it becomes apparent that a linear data representation is simply a factorization of the data matrix. Principal component analysis, independent component analysis, vector quantization, and non-negative matrix factorization can all be seen as matrix factorization, with different choices of objective function and/or constraints [22]. Finally, a variant of NMF was used to solve the extractive ATS problem [48, 59] showing good results. The mentioned NMF algorithms for ATS use W matrix as the weights for each of the topics features on matrix h. These weights are used to estimate the relevance of each sentences in the defined topics space found with h matrix. The extractive summary is obtained by selecting the most relevant sentences.

2.7 Summary

This chapter corresponds to the state of the art for Automatic Text Summarization (ATS). It introduces the basic concepts for ATS, documents representation and techniques used for generate Automatic Extractive Summaries. Also, it presents how clustering has been used for solving the ATS problem and some document clustering algorithms. The ECSAGO details are also discussed in this chapter because the proposed algorithm is an extension of it. The next chapters discuss the details of the created algorithm, their results and the further research.

3 SENCLUS Algorithm

Automatic text summarization has become a relevant topic due to the information overload. This automatization aims to help humans and machines to deal with the vast amount of text data (structured and un structured) offered in the web and deep web. Using a genetic clustering algorithm, SENCLUS clusters the sentences as close representation of the text topics using a fitness function based on redundancy and coverage, and applies a scoring function to select the most relevant sentences of each topic to be part of the extractive summary. Also, the advantages of using a clustering technique over a supervised technique is that it requires less human intervention. SENCLUS requires no specific number of topics and it is capable to automatically detect the number of topics without human intervention, which is an important advantage over other algorithms. This is possible due to the topics detection model in which SENCLUS is based.

3.1 Topics Representation

It is a fact that a writing is a representation of ideas that the writer intends to transmit. These ideas are also known as text latent topics [9]. For very small documents the number of latent topics tends to one, but for longer writings this number is larger. Besides, on every writing there is a main idea or a set of main ideas around which the text is written. Therefore, there should be a set of relevant latent topics that dominate the full text. Each cluster corresponds to a latent topic and its size (number of sentences) is the topic relevance. To select the best summary sentences SENCLUS ranks each sentence using a score value based on cluster relevance (number of sentences in the cluster) and the similarity between the sentence and the clusters centers to which it corresponds.

A text is a written representation of one or more ideas that are intended to be expressed by the writer. Each one of these ideas could be expressed by one or more sentences. To summarize a text it is necessary to detect the ideas or topics, and then select the sentences subset which is an optimal representation constrained by size.

Let us define I as the set of ideas which the writer wants to represent in the text. A property of I is that it does not change after being written and any misunderstanding of the text intention (also I) occurs due to a bad writing or bad reading.

Until now it has been established that a text is an approximated representation of one or more ideas which the writer intends to communicate; also, that the intended set of ideas ($idea_t \in I$) are susceptible to the writer's translating error and to the reader's understanding error. When a text is to be summarized, the text sentences are the written representation of the text intention which is the set of ideas embodied in the text.

$$idf(t, D) = log_{10} \frac{\mid D \mid}{\{d \in D : t \in d\}} \tag{3-1}$$

$$tf - idf\,(t, d, D) = tf\,(t, d) \times idf(t, D) \tag{3-2}$$

The way of representing text numerically has been studied by many researchers who have worked with the problem of semantics, and they conclude that the meaning of words is closely connected to the statistics of word usage [60]. The historical use of numerical vectors to represent text has showed how powerful and useful this is [60, 41]. In this case, the vector space represents the text as a $m \times n$ matrix in which the vertical axis represent the sentences and the horizontal axis represent the terms found in the text. Each sentence vector contains a numerical value with the term frequency-inverse document frequency ($tf - idf$). The $tf - idf$ definition in equation (3-2) uses the idf definition in equation (3-1) to measure how much information the term provides, where N is the total number of documents and $\{d \in D : t \in D\}$ is the total number of documents in which t appears. Differently from term frequency (tf), a big $tf - idf$ value is an indicator of term relevance. Using the vector space representation and modeling the sentences as documents and the terms as dimensions, it is possible to cluster the sentences into k groups and use clusters centroids as representation of each $idea_t \in I$. These centroids are only numerical vectors which do not represent text sentences and therefore are *hypothetical sentences*. For example, say a text is made of two sentences s_1 and s_2. To represent the sentences in the vector space, the $tf - idf$ value is used. The term frequencies table for sentence s_1 are showed in Table **3-1** and for sentence s_2 in Table **3-2**, and the terms idf are presented in Table **3-3**. Using the mentioned tables, the vector space representation for s_1 and s_2 is in Table **3-4**.

Table **3-1**: terms frequencies for s_1

Term	Term Count
this	1
is	1
a	2
sample	1

Table **3-2**: terms frequencies for s_2

Term	Term Count
this	1
is	1
another	2
example	3

Table **3-3**: terms inverse document frequencies

Term	Term Inverse Document Frequency
this	0
is	0
a	0.301
another	0.301
sample	0.301
example	0.301

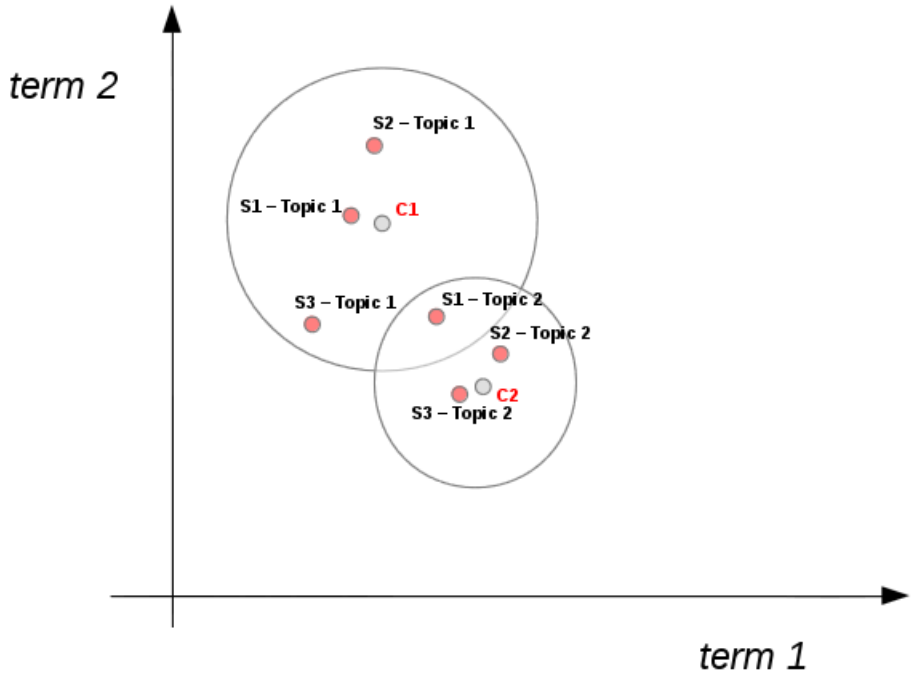
Table **3-4**: $tf - idf$ vector space representation

tf-idf	this	is	a	another	sample	example
s_1	0	0	0.602	0	0.301	0
s_2	0	0	0	0.602	0	0.903

The vector space representation in Table **3-4** is extracted from s_1 and s_2 which are sentences present in the original text. But, if a vector with the mean value of each dimension is used as cluster center, then it is clear that the calculated vector does not represent a sentence present in the original text. Then, this new vector represents a hypothetical sentence. As can be seen in Figure **3-1** where each dark point represents a sentence, there is a set of centers for each cluster $\{c_1, c_2\}$. These centers are the *hypothetical sentences* represented in the vector space.

Based on that idea, the sentences are clustered and their clusters centers are used as a good approximation of the text *hypothetical sentences* or latent topics.

Figure **3-1**: Text representation at sentence level in the Vector Space 2-D



3.2 Optimization Problem

Let us define $S = \{s_1, s_2, \ldots, s_m\}$ as the set of sentences extracted from the analyzed text and $T = \{t_1, t_2, \ldots, t_n\}$ as the text terms, which implies that each sentence vector s_i has a size n or $\mid s_i \mid = n$, where $s_i \in S$.

As mentioned, an $idea_t \in I$ is represented by one or more sentences. Therefore similar sentences must represent a part of a same $idea_t$. Then, a good sentences group that represents a topic is compound by relevant sentences which are similar to each other. Traditional intra and inter cluster measures can be used to evaluate the "goodness" of a sentences group. Because the objective is to measure the quality of a topic (sentences group), the coverage and redundancy can be used to capture intra and inter clusters measures in the "semantic space" (vector space). Coverage in (3-3) measures the relevance of a sentence and Redundancy in (3-4) measures how similar or compact a group of sentences is.

$$coverage(s_i) = \sum sim(s_i, S) \tag{3-3}$$

$$redundancy(s_j) = \sum_{s_k \in ss} sim(s_j, s_k) \tag{3-4}$$

Coverage presented in equation (3-3) models the relevance of a sentence s_i in the text. The $\arg_i \max(coverage(s_i))$ will be such that s_i which fulfills the condition $\sum_{s_j \in S} sim(s_i, s_j) = 1$ always TRUE being 1 the maximum value for $sim(s_i, s_j)$. Then, the higher is the coverage the better representation is the sentence of the analyzed text. On the other hand, **Redundancy** presented in equation (3-4) models how much a sentence s_j belongs to a topic represented by a sentences subset $ss_x \in S$. And, the same as with the coverage, the higher is the redundancy of s_j the better s_j is a representation of the sentences subset $ss_x \in S$. Finally, if $\mid ss_x \mid < \mid S \mid$ then $redundancy(s_i) < coverage(s_i)$.

The clustering problem is an optimization problem that tries to maximize the intra-cluster measure and minimize the inter-cluster measure. The proposed objective function (3-5) maximizes the average redundancy and minimizes the average coverage. The average coverage is multiplied by (-1) to transform the problem into a maximization problem. The complete objective function to be optimized is defined in equation (3-5), where k is the expected number of topics..

$$\arg\max_{s_i, s_j, \ldots} f(s_i, s_j, \ldots) = \sum_{x=1}^{k} h(ss_x) \tag{3-5}$$

$$h(ss_x) = \sum_{s_i \in ss_x} \left(\frac{redundancy(s_i)}{\mid ss_x \mid} \right) - \left(\frac{coverage(s_i)}{\mid S \mid} \right) \tag{3-6}$$

The function $f(s_i, s_j, \ldots)$ will be maximized to find the set of hypothetical sentences or cluster centers $s_i, s_j, \ldots$ that maximize the $h(ss_x)$ of each cluster or group. And theoretically $h(ss_x)$ reach their maximum when all sentences in ss_x belong to the same topic.

To measure if two sentences $s_i, s_j \in S$ talk about a similar topic, the cosine similarity and the extended Jaccard similarity are used. The cosine similarity between two vectors defined in (3-7) and the extended Jaccard measure or Tanimoto coefficient is defined in (3-8).

$$sim(s_i, s_j) = \frac{\sum_{x=1}^{n} s_{ix} \times s_{jx}}{\sqrt{\sum_{x=1}^{n} s_{ix}^2} \sqrt{\sum_{x=1}^{n} s_{jx}^2}} \tag{3-7}$$

$$sim(s_i, s_j) = \frac{\sum_{x=1}^{n} s_{ix} \times s_{jx}}{\sum_{x=1}^{n} s_{ix} + \sum_{x=1}^{n} s_{jx} - (\sum_{x=1}^{n} s_{ix} \times s_{jx})} \tag{3-8}$$

The extended Jaccard coefficient can be used for handling the similarity of documents data in text mining. In the case of binary attributes, it reduces to the Jaccard coefficient. For text documents, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. So, the extended Jaccard could be a better similarity measure than the cosine measure because extended Jaccard also takes into account what the two vectors do not have in common.

3.3 Proposed Algorithm

The proposed objective function presented in equation (3-5) could be solved modelling it as a mutimodal optimization problem in which each cluster is maximized using the equation (3-6). Being now a multimodal problem, it can be solved using a Genetic Algorithm (GA) or niching strategies.

ECSAGO is a genetic clustering algorithm which is robust to noise and has the ability to detect the number of clusters automatically using niching. The advantage of genetic algorithms over other methods is that, with a good set of genetic operators, a good solution could be found in a time t; and t depends on the termination criteria for the algorithm, configured at the beginning. Also, genetic operators like selection, mutation and crossover allow to explore the function landscape and refine the promissory areas until finding the local or global optimal .

ECSAGO has been used for document clustering showing good results [28], but SENCLUS takes all ECSAGO advantages to solve the proposed objective function defined in (3-6) which is not density based as the ECSAGO fitness function. The ECSAGO fitness function is the density of the hypothetical cluster which is completely different to SENCLUS fitness function based on redundancy and coverage. Alsom because it's fitness function is not density based, SENCLUS radius is different and it is used as a topic border. These were the reasons to create SENCLUS.

SENCLUS keeps the concept of a dense clusters that represents topics along with Deterministic Crowding, restricted mating and the HAEA to adapt the relevance of each operator to decide about the frequency with which it should be used. SENCLUS adopts a radius used to model the topics boundaries in the vector space representation.

After the sentences were clustered, the clustering results are analyzed. A relevance function is used to give a score to each sentence, and by this score the sentences will be ranked. The $score(s_j)$ calculates the similarity between the sentence and each cluster center $sim(s_j, c_i)$.

Algorithm 3.1 SENCLUS pseudo code

Calculate coverage for each sentence in S
Select random sentences as initial population
Assign the $radius_{initial}$ to all the initial population
WHILE $generation < maxGenerations$:
  FOR $individual$ IN $population$:
    $individual_{fitness} = calculateFitness(individual_{vector}, individual_{radius}, population)$
  $parents = generateCouples(population)$
  FOR $parentsCouple$ in $parents$:
    IF $restrictedMating(parentsCouple)$:
      $children = applyOperatorHAEAwithCrossover(parentsCouple)$
    ELSE
      $children = applyOperatorHAEA(parentsCouple)$
    FOR $child$ IN $children$:
      $child_{radius} = updateSigma(child_{radius}, child_{vector}, population)$
    $winners = deterministicCrowding(children, parentsCouple)$
    $replace(parentsCouple, winners, population)$
$sentencesScoring(population, S)$

Algorithm 3.2 sentences scoring

FOR $sentence$ IN S:
  FOR $individual$ in $population$:
    IF $similarity(sentence, ind) > individual_{radius}$:
      $sentence_{clusters} = concat(sentence_{clusters}, individual_{id})$
FOR $sentence$ IN S:
  FOR $clusterCenter$ IN $sentence_{clusters}$:
    $sentence_{score} = similarity(sentence, clusterCenter) * (\frac{1}{sentence_{textPosition}})$
$sort(sentences_{score})$

Algorithm 3.1 shows the SENCLUS pseudo code. The algorithm starts by selecting p sentences randomly, being p the population size. After that, the evolutionary process starts by calculating the fitness of each individual and generating their offspring taking into account the mating restriction. This mating restriction keeps the niches by crossing only individuals that belong to the same niche. At the end of each generation, the Deterministic Crowding (DC) is used to decide which individuals survive to be part of the next generation by selecting the ones that were better than their parents. Finally, after the evolutionary process has ended a scoring function is applied to rank the sentences by their relevance in the text based on the detected topics. The scoring function showed in Algorithm 3.2 give more relevance to a sentence that is a close representation of the topic which is also a relevant topic. This rank allows to select the best sentences that are going to be in the extract.

3.3.1 Representation

Each individual represents a potential *hypothetical sentences* that represents a latent topic. These individuals are initialized randomly selecting vector representations of sentences present in the text, using sentences as documents and terms as dimensions. Each individual has a length n, where n is the number of terms present in the text, and each gene is a float number representing the term relevance.

3.3.2 Genetic Operators

One Dimension Linear Crossover Applies a linear crossover to a single component.

$$parent_{a,x} = parent_{a,x} * \beta + parent_{b,x} * (1 - \beta)$$

$$parent_{b,x} = parent_{a,x} * (1 - \beta) + parent_{b,x} * \beta$$

One Dimension Simple Crossover Exchanges one component of the first individual with the same component of the second individual.

One-point Crossover A single crossover point on both parents organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children.

Two-point Crossover Two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

Heuristic Crossover A crossover operator that uses the fitness values of the two parent chromosomes to determine the direction of the search. The are created according to the following equations:

$$child_{a,x} = \beta(Parent_{best,x} - Parent_{worst,x}) + Parent_{best,x}$$

$$child_{b,x} = \beta Parent_{best,x} + (1 - \beta)Parent_{worst,x}$$

$$0 \leq \beta \leq 1 \text{ random}$$

Random Mutation It is analogous to biological mutation. Mutation alters one gene value in a chromosome from its initial state randomly by adding a percentage of the initial value. Another random variable is used to decide if the added value is positive or negative.

Gaussian Mutation Changes one component of the encoded real vector with a number randomly generated following a Gaussian distribution using as mean the old value of the component, and the given standard deviation.

3.3.3 Fitness Function

The fitness value for the j^{th} candidate center c_j, is defined using the function :

$$f(c_j) = redundancy(c_j) - coverage(c_j) \tag{3-9}$$

,where S is the set of sentences extracted from the text, redundancy is (3-4) and coverage (3-3).

The fitness value of each individual requires a radius to allocating sentences in the groups using the function defined in (3-10). Also, radius allow soft clustering and it delimits each cluster for the Deterministic Crowding. The cluster radius represents the topic scope in the vector space. SENCLUS decides whether a sentence belongs or not to a cluster using the condition defined in(3-10), where c_j is a hypothetical sentence or cluster center.

$$IF\ sim(s_i, c_j) > radius \quad THEN\ s_i \in c_j \tag{3-10}$$

The radius is updated with the mean difference between the similarity of each sentence against the cluster center and the coverage of the sentence. The radius will reach their maximum when the cluster sentences belong to only one topic with a high confidence represented with a good sentence coverage and a high similarity between cluster sentences and cluster center.

The candidate center c_j $radius(c_j)$ is defined in (3-11).

$$radius(c_j)_t = radius(c_j)_{t-1} + \frac{\sum_{i \in c_j} sim(s_i, c_j) - \frac{sim(s_i, S)}{|S|}}{\mid c_j \mid} \tag{3-11}$$

3.3.4 Sentences Scoring and Selection

This scoring function exists because an extractive summary could not be formed by *hypothetical sentences* which are float vectors, so a set of the best sentences should be selected. After the sentences have their score, they are sorted and added from the top to the bottom, until there is no space in the summary.

The sentence scoring function is defined in (3-12).

$$score(s_j) = sim(s_j, c_{i,s_j \in c_i}) \times \mid c_i \mid \times \left(\frac{1}{pos(s_j)} \right) \tag{3-12}$$

Finally the best r sentences are selected, where r depends on the summary length. The pseudo code is shown in Algorithm 3.2.

3.4 Summary

This chapter introduced the SENCLUS algorithm. This algorithm was inspired on the ECSAGO algorithm and was specially designed to generate extractive summaries using sentences clustering. SENCLUS use a fitness function based on redundancy and coverage. To maintain the detected clusters, SENCLUS uses the $radius(c_i)$ of the cluster c_i. Finally, the performed experiments are presented and discussed in the next chapter.

4 Experiments and Results

4.1 DUC 2002 Data Set

The DUC 2002 data set provided by the Document Understanding Conference [12], is a data set prepared for testing task of single and multiple document summarization. The documents of the DUC 2002 collection are categorized in subgroups and each subgroup has a set of control summaries which were generated by experts.

For single document summaries, the generated extracts summaries have a maximum of 200 words. The DUC 2002 composition details are described in Table **4-1**.

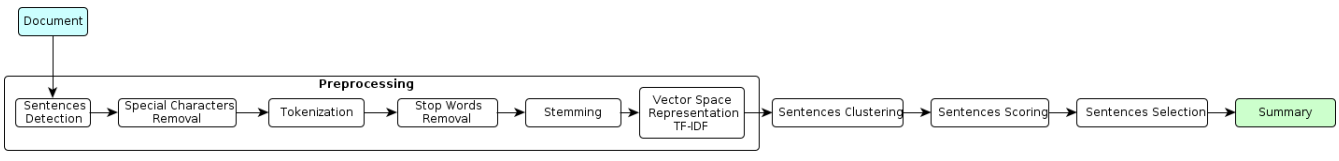
Table **4-1**: DUC 2002 Details

	DUC 2002
number of document collections	59
number of documents in each collection	10
data source	TREC
summary length	200 words

4.2 Preprocessing

Before apply the algorithm the text is parsed to extract the sentences, removing the special characters of the sentences, and then represent the sentences using the vector space model removing stops words and applying stemming to words. The overall pipeline can be seen in Figure **4-1**.

Figure **4-1**: Pipeline Design



4.3 Experiments

As mentioned in Chapter 2, there are few extractive ATS algorithms that use clustering or sentences clustering. Then, three clustering algorithms were selected to be compared against SENCLUS. The selected algorithms were K-means, GK-means and NMF. The reasons to select K-means were that it is a well known algorithm which most of the time is the first choice to solve any clustering problem and a K-means sentences clustering algorithm for extractive ATS has reported interesting results [68]. Then, this algorithm could be the base bottom from where SENCLUS could be compared against other clustering algorithms. Equally important is their genetic variance GK-means which was designed to converge to global optimum. And finally, NMF is one of the most robust algorithms use for classification and clustering. NMF use a matrix decomposition to model hidden patterns and a variant of it was used to solve the extractive ATS problem showing good results [48, 59]. The three algorithms were implement to solve a generic clustering problem and all of them use the same processed data set use by SENCLUS. Each algorithm (SENCLUS included) use the same data set as input to cluster the sentences and apply the same sentences scoring function showed in (3-12) to rank the sentences. Finally, the extractive summary is generated selecting the best sentences using the rank until there is no more space in the summary constrained by a maximum length of 200 words.

The results of each experiment are listed bellow.

4.3.1 K-means experiments

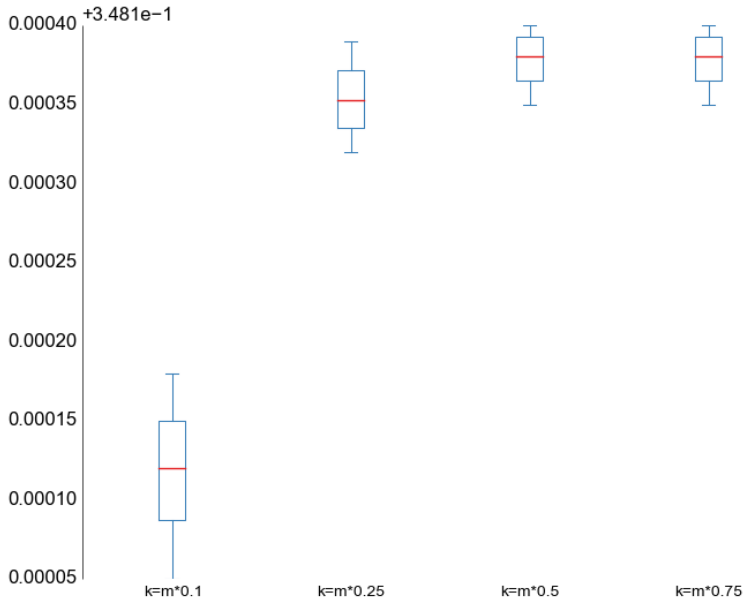
K-means algorithm has 2 variable parameters. These are k which for this experiments is expected number of latent topics and the maximum number of iterations which is used as termination condition. The values set used for each parameter are listed in Table **4-2**. Each possible configuration was run 1000 times to obtain a representative sample.

Table **4-2**: k-means configurations

Parameter	Values
k	$m * 0.1, m * 0.25, m * 0.5, m * 0.75$
iterations	$10, 25, 50, 100, 150, 250, 400$

The results showed that the variation of K has an impact on the algorithm performance. The Figure **4-2** show a summary of the algorithm behavior when K varies. The figure shows k (number of clusters) affects the results but the effect is small because the algorithm converges into values between 0.381. On the other hand, the impact of use different number of iterations is small and the results behaved as expected returning better results with a bigger number of iterations, but the difference was also small among them.

Figure **4-2**: K-means behavior varying K



The best K-means algorithm results are listed in Table (**4-3**).

Table **4-3**: K-means best experiments results

Iterations	K	Rouge-1	Rouge-2
400	$m * 0.1$	0.3482	0.1267
400	$m * 0.25$	0.3485	0.1300
400	$m * 0.5$	0.3485	0.1300
400	$m * 0.75$	0.3485	0.1300

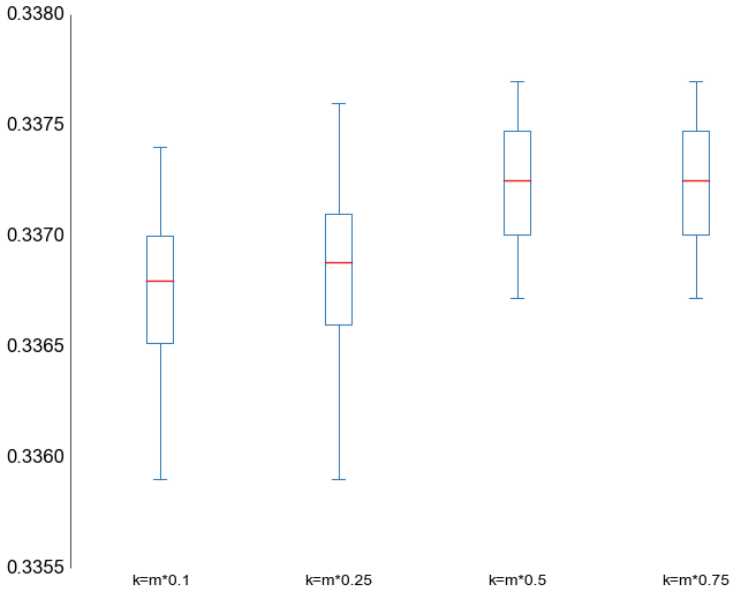
4.3.2 GK-means experiments

GK-means algorithm has 2 variable parameters. These are k which for this experiments is expected number of latent topics and the maximum number of iterations which is used as termination condition. The values set used for each parameter are listed in Table **4-4**. Each possible configuration was run 1000 times to obtain a representative sample.

Table **4-4**: GK-means configurations

Parameter	Values
k	$m * 0.1, m * 0.25, m * 0.5, m * 0.75$
iterations	$10, 25, 50, 100, 150, 250, 400$

Figure **4-3**: GK-means behavior varying K



The results showed that the variation of K has an impact on the algorithm performance. The Figure **4-3** show a summary of the algorithm behavior when K varies. The figure shows k (number of clusters) affects the results generating better summaries with populations between $[m * 0.5, m * .75]$. On the other hand, the impact of use different number of iterations is small and the results behaved as expected returning better results with a bigger number of iterations, but the difference was small among them.

The best GK-means algorithm results are listed in Table (**4-5**).

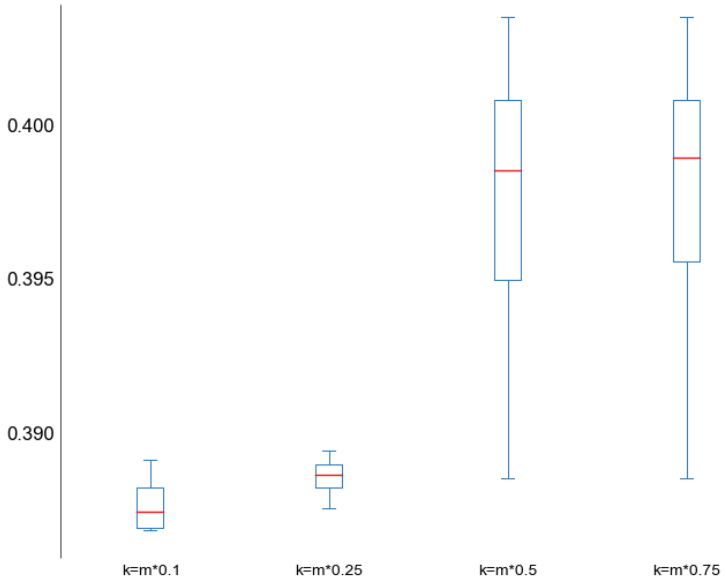
Table **4-5**: GK-means best experiments results

Iterations	K	Rouge-1	Rouge-2
400	$m * 0.1$	0.3374	0.1290
400	$m * 0.25$	0.3376	0.1291
400	$m * 0.5$	0.3377	0.1301
400	$m * 0.75$	0.3377	0.1301

4.3.3 NMF Experiments

NMF algorithm has 2 variable parameters. These are k which for this experiments is expected number of latent topics and the maximum number of iterations which is used as termination

Figure **4-4**: NMF behavior varying K



condition. The values set used for each parameter are listed in Table **4-6**. Each possible configuration was run 1000 times to obtain a representative sample.

Table **4-6**: NMF configurations

Parameter	Values
k	$m * 0.1, m * 0.25, m * 0.5, m * 0.75$
iterations	$10, 25, 50, 100, 150, 250, 400$

The results showed that the variation of K has an impact on the algorithm performance. The Figure **4-4** show a summary of the algorithm behavior when K varies. The figure shows k (number of clusters) affects the results generating better summaries with populations between $[m * 0.5, m * .75]$. On the other hand, the impact of use different number of iterations is small and the results behaved as expected returning better results with a bigger number of iterations, but the difference was small among them.

The best NMF algorithm results are listed in Table (**4-7**).

Table **4-7**: NMF best experiments results

Iterations	K	Rouge-1	Rouge-2
400	$m * 0.1$	0.386	0.1460
400	$m * 0.25$	0.4035	0.1611
400	$m * 0.5$	0.4036	0.1611
400	$m * 0.75$	0.4036	0.1611

4.3.4 SENCLUS Experiments

In this section the reports of all the executed experiments are presented. The algorithm use three parameters which are population size, generations and initial radius; and can use different genetic operators to find the solution. For each parameter a domain is defined and the experiments are executed over the whole set of possible combination given those domains. The parameters domains are showed in Table **4-8**.

Table **4-8**: used parameters values

Parameter	Values
population size	$m * 0.1, m * 0.25, m * 0.5, m * 0.75$
generations	$10, 25, 50, 100, 150, 250, 400$
initial radius	$0.1, 0.01, 0.001, 0.0001, 0.00001$

The mentioned parameters domains implies that there are 140 possible combinations without taking into account the genetic operators. Also, depending on the used genetic operators the results could vary, so a set of combined genetic operators were created to test the performance of each set of genetic operators. The genetic operators set is listed in Table **4-9**.

Each set of genetic operators listed in Table **4-9** were tested using each one of the possible different parameters combinations listed in Table **4-8**. The best results of those experiments are listed in Table **4-10**.

Additionally, the Figure (**4-5**) shows a summary of how SENCLUS behaves using different population size, Figure (**4-7**) show the behavior with after using different initialization values for radius and Figure (**4-6**) show the effect of use more or less iterations. The Figure (**4-5**) shows that the population size has an important effect on the algorithm performance and that the best results are obtained with values $[m * 0.5, m * 075]$. The Figure (**4-6**) and Figure (**4-7**) show that the iterations and the initial sigma also impact the algorithm performance. In the case of iterations, the algorithm converges after 150 iterations and their results do not change at all. Also, it is important to remark that the results obtained with iterations between $[50, 100]$ are not to far from the results obtained with 150 generation, making a range between $[50, 150]$ a good iterations range. Finally, the initial radius has an impact

Figure **4-5**: SENCLUS behavior varing population size

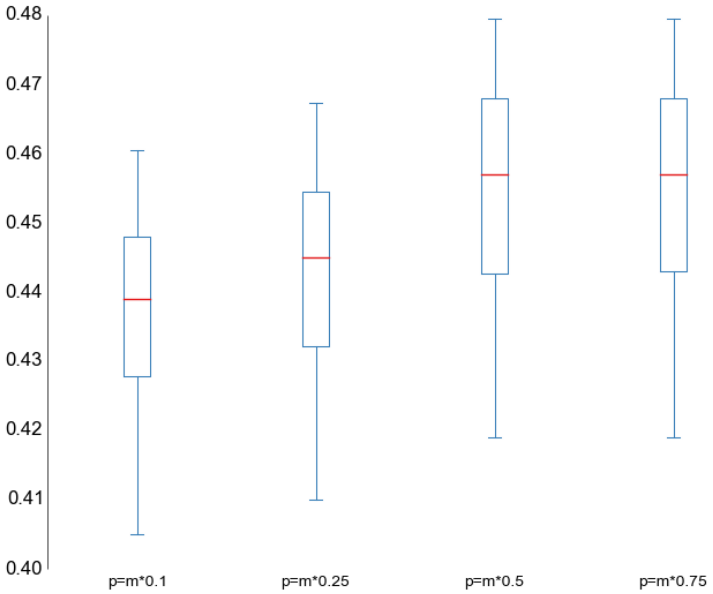


Figure **4-6**: SENCLUS behavior varing iterations

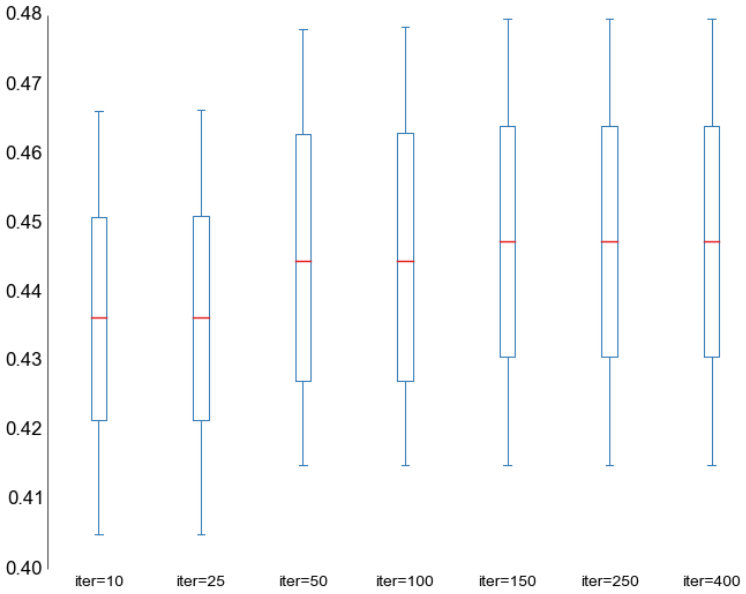


Table **4-9**: first set of applied genetic operators

Operators Sets
random mutation
gaussian mutation
heuristic crossover
two point crossover
one point crossover
one dimension simple crossover
one dimension linear crossover
random mutation, heuristic crossover
random mutation, two point crossover
random mutation, one dimension simple crossover
random mutation, one dimension linear crossover
gaussian mutation, heuristic crossover
gaussian mutation, two point crossover
gaussian mutation, one dimension simple crossover
gaussian mutation, one dimension linear crossover

on the algorithm when the initial radius is to big. The reason is that after analyze the experiments results it was clear that if the initial radius is too big, it can not be adapted properly. On the other hand, if the initial radius is too small there is no problem because after some iterations the radius adapts himself properly.

From the first set of experiments listed in Table **4-10**, it can be seen that even with only one operator the algorithm return good results with $Rouge-1$ values above of 0.410. Furthermore, the results using only one genetic operator tell us that the crossover could be more relevant than mutation to find an optimal solution. This could be explained with the fact that the used expected number of topics is high and because of this a exploitation operator is more useful for SENCLUS than an exploration operator. Based on the results showed in Table **4-10**, the genetic operators with the best behaviors were selected to apply them in groups of 3 operators. The new operator set is described in Table **4-11**.

The results of the experiments using sets of 3 operators showed in Table **4-12** outperforms the overall results. The reason for this could be that with a good combination of exploration and exploitation the algorithm could find better cluster and therefore improve the results. The fact that the best result use the heuristic crossover operator could be explained by the use of the fitness value on this operator to the generate an offspring. This strong relation with the fitness function direct the operator with the more promissory zones of the analyzed space section.

Table **4-10**: best results per operators set

Operators Sets	Parameters	Cosine		Jaccard	
		Rouge-1	Rouge-2	Rouge-1	Rouge-2
random mutation	iter=150 radius=0.00001 pop=m/2	0.4233	0.190	0.4235	0.190
gaussian mutation	iter=150 radius=0.00001 pop=m/2	0.419	0.183	0.420	0.183
heuristic crossover	iter=150 radius=0.00001 pop=m/2	0.436	0.209	0.436	0.210
two point crossover	iter=150 radius=0.00001 pop=m/2	0.430	0.211	0.431	0.211
one point crossover	iter=150 radius=0.00001 pop=m/2	0.429	0.211	0.427	0.209
one dimension simple crossover	iter=150 radius=0.00001 pop=m/2	0.420	0.14	0.420	0.14
one dimension linear crossover	iter=150 radius=0.00001 pop=m/2	0.406	0.12	0.406	0.12
random mutation, heuristic crossover	iter=150 radius=0.00001 pop=m/2	0.473	0.217	0.473	0.217
random mutation, two point crossover	iter=150 radius=0.00001 pop=m/2	0.471	0.214	0.471	0.214
random mutation, one point crossover	iter=150 radius=0.00001 pop=m/2	0.471	0.216	0.471	0.216
random mutation, one dimension simple crossover	iter=150 radius=0.00001 pop=m/2	0.448	0.173	0.447	0.171
random mutation, one dimension linear crossover	iter=150 radius=0.00001 pop=m/2	0.469	0.211	0.469	0.211
gaussian mutation, heuristic crossover	iter=150 radius=0.00001 pop=m/2	0.4722	0.213	0.4721	0.213
gaussian mutation, two point crossover	iter=150 radius=0.00001 pop=m/2	0.469	0.213	0.469	0.213
gaussian mutation, one point crossover	iter=150 radius=0.00001 pop=m/2f	0.468	0.211	0.468	0.211
gaussian mutation, one dimension simple crossover	iter=150 radius=0.00001 pop=m/2	0.438	0.168	0.438	0.168
gaussian mutation, one dimension linear crossover	iter=150 radius=0.00001 pop=m/2	0.436	0.162	0.436	0.162

Figure **4-7**: SENCLUS behavior varing radius

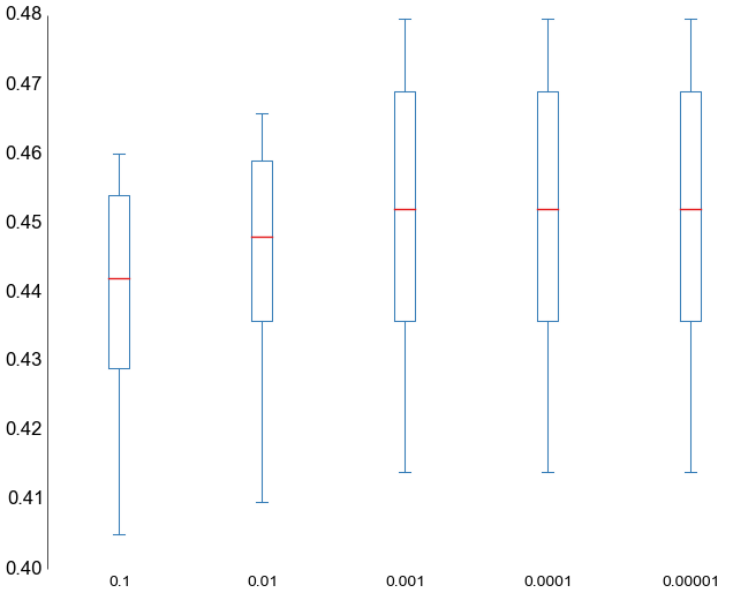


Table **4-11**: second set of applied genetic operators

Operators Sets
one point crossover, heuristic crossover, random mutation
one point crossover, heuristic crossover, gaussian mutation
two point crossover, heuristic crossover, random mutation
two point crossover, heuristic crossover, gaussian mutation

Table **4-12**: second set best results per operators sets

Operators Sets	Params	Rouge-1	Rouge-2
one point crossover, heuristic crossover, random mutation	iter=150 radius=0.00001 pop=m/2	0.4795	0.2200
one point crossover, heuristic crossover, gaussian mutation	iter=150 radius=0.00001 pop=m/2	0.473	0.2173
two point crossover, heuristic crossover, random mutation	iter=150 radius=0.00001 pop=m/2	0.4722	0.213
two point crossover, heuristic crossover, gaussian mutation	iter=150 radius=0.00001 pop=m/2	0.4726	0.214

4.4 Discussion

The experiments results reported in subsections 4.3.1,4.3.2,4.3.3 and 4.3.4 showed that SENCLUS clearly outperforms K-means, GK-means and NMF. The objective of this comparison was to see if the designed algorithm will perform better or worst than a clustering algorithm designed to solve clustering problems. Besides, during the research some variations of NMF [48, 59] for the extractive ATS problem was found, and SENCLUS also overcome the reported results by these variations. The explanation for these fact is that SENCLUS was specifically designed to solve the extractive ATS and their fitness function and radius provide a great advantage to the algorithm over clustering algorithm which are not tuned for an specific problem. Then, the Table **4-13** compares the proposed genetic clustering algorithm against other algorithms over their reported results using the DUC2002 data set and ROUGE. The SENCLUS perform well compared against the best state of the algorithm and the reported results are not too far from the best ones.

Table **4-13**: DUC2002 results

Algorithm	Rouge-1	Rouge-2
UnifiedRank[63]	0.4847	0.2146
MA-MultiSumm[37]	0.4828	0.2284
SENCLUS	**0.4795**	**0.2200**
DE[5]	0.4699	0.1236
FEOM[55]	0.4657	0.1249
NMF	0.4036	0.1611
K-means	0.3485	0.1300
GK-means	0.3377	0.1301

It is curious fact that the first three (3) algorithms (UnifiedRank, MA-MultiSumm and SENCLUS) use the concept of global relevance and local relevance to find the best extractive summary but each one apply it differently. Even though Unified Rank and MA-MultiSumm are better than SENCLUS they are not too far from the best SENCLUS results. Besides, SENCLUS has the advantage that his summarization model and equations are simpler than the other ones which offers the possibility of improvement to the algorithm. Indeed, SENCLUS could take advantage of some aspects of UnifiedRank and MA-MultiSumm to boost the fitness function and extract better clusters.

The text in Figure **4-8** is a document from DUC 2002 and the content in Figure **4-9** is the SENCLUS generated summary. The extract summary showed in Figure **4-9** is one of the best extractive summaries generated by the algorithm. A manual verification of the results indicates that in some cases the algorithm is not capable to generate good summaries because some documents have irregular structures which made impossible parse those documents correctly in a automatic way. This issue is strongly related with the structure of DUC 2002 documents which uses tags to name document sections. Then, it is possible to get better results with a cleaner data set, but in real problems is hard to find a clean data set therefore is better to report the results obtained from the original documents without manual modifications.

4.5 Summary

This chapter presented the results obtained after apply the K-means, GK-means, NMF and SENCLUS to generate single document extractive summaries using the DUC2002 dataset.

Figure **4-8**: Text Example

Text
TITLE:President Clinton, John Major Emphasize 'Special Relationship'. **Article Type:BFN [Text] Washington, February 28 (XINHUA) – U.S. President Bill Clinton, trying to brush aside recent differences with London, today stressed Washington's special transatlantic relationship with Britain. Welcoming British Prime Minister John Major in Pittsburgh, where major's grandfather and father once lived, Clinton said at the airport, "We're working together today to respond to the terrible tragedy in Bosnia to try to bring an end to the killing and to bring peace and to keep that conflict from spreading." For his part, Major said, pressure would be increased for the peace that every sensitive person wishes to see in that war-torn and troubled land. On Russia, Major said "A Russia that's a good neighbor to the United States and West would be one of the finest things that this generation could hand down to the next." Clinton will then share his Air Force One back to the nation's capital. Major will spend a night at the White House, the first foreign head of state to have this honor since Clinton became President. On Tuesday [1 March], the two leaders will begin their discussions on a wide range of issues including Russia, Bosnia, Northern Ireland and the world trade. The two will also discuss Northern Ireland and "what to do with NATO," Clinton said. Clinton and major will meet again in June in Europe during the commemoration of the 50th anniversary of D-Day of the second world war. Major said Clinton would visit Britain, and perhaps the Oxford University, Clinton's alma mater, during the June visit.**

Figure **4-9**: Extract Summary

Summary
Welcoming British Prime Minister John Major in Pittsburgh, where majors grandfather and father once lived, Clinton said at the airport, "We're working together today to respond to the terrible tragedy in Bosnia to try to bring and end to the killing and to bring peace and to keep that conflict from spreading". On Rusia, Major said "A Russia that's a good neighbor to the United States and West would be one of the finest things that this generation could hand down to the next". February 28 (XINHUA) - U.S President Bill Clinton, trying to brush aside recent differences with London, today stressed Washington's special transatlantic relationship with britain.

The reported results showed that SENCLUS outperforms K-means, GK-means, NMF and other state of the art algorithms, but it can not still outrival the best two (2) state of art algorithms; although SENCLUS is really close to the best two (2) state of art algorithms.

5 Conclusions and Further Research

5.1 Conclusions

In this work, a new genetic clustering algorithm for single document text summarization called SENCLUS was presented. This algorithm uses sentences clustering to detect the text topics. SENCLUS is an extension of ECSAGO and it uses a fitness function based on redundancy and coverage together with a radius which represents the topic coverage. This algorithm evolves a population of individuals in which each individual is a cluster center. The mentioned fitness function maximizes the inter cluster measure which is represented by the redudancy function, and minimizes the intra cluster measures which is presented by the coverage function. The fact that fitness function was defined in this way is not new and has been used to solve other clustering problems with the differences that the inter and intra clusters measuring functions could be defined differently for other problems. Then, it can be concluded that the proposed fitness function was properly defined, with the concern that the performance of this fitness function also depends on the similirity function used to measure how similar are two objects, or for the ATS problem, how similar is a pair of sentences.

SENCLUS uses a cluster radius to delimit each cluster which allows soft clustering with which is possible that one sentence could belong to one or more topics. This algorithm aspect offers an important advantage. This flexibility allows sentences to take advantage of context by letting two or more sentences cooperate for representing one or more topics at a time. Other innovative features of SENCLUS are: a topics detection model using sentences clustering, the use of radius to delimit a topic in the vector space, the summarization model, and all the advantages offered by the ECSAGO like the Deterministic Crowding (DC) and HAEA to solve clustering problems. All these innovations make this algorithm a robust technique which is capable to detect the number of topics automatically without being susceptible to noisy document sentences which are sentences that do not belong to the document main topic stream and they look like spam sentences.

Finally, the conducted experiments presented in this document reveals that SENCLUS is a promissory technique for the single document text summarization problem which could be improved updating the fitness function or using other genetic operators. From the experiments results obtained, SENCLUS demonstrates a much better performance than K-means, GK-means and NMF generating extractive summaries, with the advantage that it does not

require the number of clusters as the three (3) mentioned clustering algorithms do. Also, the algorithm is in the top three (3) algorithms in state of the art that were used for single document extractive text summarization using the DUC2002 dataset. This fact reinforces the conclusion that the proposed approach is valuable and promissory. The experiments also concluded that use the cosine similarity or the extended Jaccard similarity makes almost no difference in the result, making them interchangeable.

When this research started it was clear that the task of generate automatic summaries was hard, but at that moment it was impossible to fully understand how challenging it could be . The hardest aspect is give meaning to text which is not easy for humans and therefore it is not going to be easier for computers. The field of text semantics and text understanding is an active field in which a lot of researches are designing better methods to measure semantic similarity between two portions of text. But, these new methods are not capable to reproduce the human precision yet and they also have a great impact on the ATS field because they are needed to find semantically similar sentences. This add even more complexity to the automatic generation of summaries. Additionally, the task of validate summaries could be considered a open problem because even today there is not a consensus about how a summary should be validated, adding more complexity by reason of which is harder to do it automatically. Now, to those problems add the challenge of natural language generation for abstract summaries and it is clear how complex is this research field. This field not only suffers from the complexities of the Natural Language Processing, it also has to deal with the complexities of solving an optimization problem as complex it is finding the best summary.

5.2 Further research

During this research it has been noticed that, the algorithm could improved in several ways and it also could be used for the multi-document extractive summarization problem with no changes. This may be proposed as further research. The use of other generic operators and a new fitness function that takes into account other factors could improve the generated extractive summaries. This new fitness function could be updated with the purpose of taking into account other factors as sentences position, similarity among sentences and the text title, among others. Furthermore, the use of another sentences similarity measure capable of finding similar semantic meaning could also boost the results. Besides, the proposed approach for topics detection constitutes an interesting modeling of the text summarization problem that could be developed to solve multi-document extractive summarization. Because the main objective is to detect the topics by clustering the sentences around them to give relevance to those sentences, for a multi-document text summarization it, could be expected more separated clusters for texts talking about different topics. Therefore the problem could be solved in a similar way as the single document problem.

Bibliography

[1] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Makrufa S. Hajirahimova. *Expert Systems with Applications*, 39(16):12460 – 12473, 2012.

[2] Rasim M. Alguliev, Ramiz M. Aliguliyev, Makrufa S. Hajirahimova, and Chingiz A. Mehdiyev. Mcmr: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications*, 38(12):14514 – 14522, 2011.

[3] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R. Isazade. Desamc+docsum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization. *Knowledge-Based Systems*, 36(0):21 – 38, 2012.

[4] Rasim M. Alguliev, Ramiz M. Aliguliyev, and Chingiz A. Mehdiyev. Sentence selection for generic document summarization using an adaptive differential evolution algorithm. *Swarm and Evolutionary Computation*, 1(4):213 – 222, 2011.

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

[6] Enrique Amigó, Julio Gonzalo, Anselmo Penas, and Felisa Verdejo. Qarla: a framework for the evaluation of text summarization systems. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 280–289. Association for Computational Linguistics, 2005.

[7] Chinatsu Aone, Mary Ellen Okurowski, and James Gorlinsky. Trainable, scalable summarization using robust nlp and machine learning. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 62–66. Association for Computational Linguistics, 1998.

[8] Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali. Fuzzy swarm based text summarization 1.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[10] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[11] Stephen Clark. Vector space models of lexical meaning. 2012.

[12] Document Understading Conference. Duc 2002 data set description and conference guidelines. http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html, 2002.

[13] John M Conroy and Dianne P O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407. ACM, 2001.

[14] Pooya Khosraviyan Dehkordi, Dr. Farshad Kumarci, and Dr. Hamid Khosravi. 57 text summarization based on genetic programming.

[15] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[16] 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.

[17] Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 57–60. Association for Computational Linguistics, 2007.

[18] George Giannakopoulos, Vangelis Karkaletsis, and George Vouros. Testing the use of n-gram graphs in summarization sub-tasks. 2008.

[19] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5, 2008.

[20] 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.

[21] EH Hovy and CY Lin. Automated multilingual text summarization and its evaluation. Technical report, Technical report Information Sciences Institute, University of Southern California, 1999.

[22] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[23] Karen . Jones. Automatic summarizing: factors and directions. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in automatic text summarization*, chapter 1. The MIT Press, 1999.

[24] Rahul Katragadda. Gems: generative modeling for evaluation of summaries. In *Computational Linguistics and Intelligent Text Processing*, pages 724–735. Springer, 2010.

[25] A. Kiani and M.R. Akbarzadeh. Automatic text summarization using hybrid fuzzy ga-gp. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 977–983, 2006.

[26] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[27] Elizabeth Leon. *Scalable and adaptive evolutionary clustering for noisy and dynamic data*. University of Louisville, 2005.

[28] Elizabeth León, Jonatan Gómez, and Olfa Nasraoui. A genetic niching algorithm with self-adaptating operator rates for document clustering. In *Eighth Latin American Web Congress, LA-WEB 2012, Cartagena de Indias, Colombia, October 25-27, 2012*, pages 79–86, 2012.

[29] Elizabeth León, Olfa Nasraoui, and Jonatan Gómez. ECSAGO: evolutionary clustering with self adaptive genetic operators. In *IEEE International Conference on Evolutionary Computation, CEC 2006, part of WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006*, pages 1768–1775, 2006.

[30] Todd A. Letsche and Michael W. Berry. *Large-Scale Information Retrieval with Latent Semantic Indexing*. 1996.

[31] Sujian Li, You Ouyang, Wei Wang, and Bin Sun. Multi-document summarization using support vector regression. In *Proceedings of DUC*. Citeseer, 2007.

[32] Marina Litvak, Mark Last, and Menahem Friedman. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 927–936, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[33] Elena Lloret and Manuel Palomar. A gradual combination of features for building automatic summarisation systems. In *Text, Speech and Dialogue*, pages 16–23. Springer, 2009.

[34] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

[35] Inderjeet Mani and Mark T Maybury. *Advances in automatic text summarization*, volume 293. MIT Press, 1999.

[36] Victoria McCargar. Statistical approaches to automatic text summarization. *Bulletin of the american society for information science and technology*, 30(4):21–25, 2004.

[37] Martha Mendoza, Susana Bonilla, Clara Noguera, Carlos Cobos, and Elizabeth León. Extractive single-document summarization based on genetic operators and guided local search. *Expert Syst. Appl.*, 41(9):4158–4169, 2014.

[38] Martha Mendoza, Carlos Cobos, Elizabeth León Guzman, Manuel Lozano, Francisco J. Rodríguez, and Enrique Herrera-Viedma. A new memetic algorithm for multi-document summarization based on CHC algorithm and greedy search. In *Human-Inspired Computing and Its Applications - 13th Mexican International Conference on Artificial Intelligence, MICAI 2014, Tuxtla Gutiérrez, Mexico, November 16-22, 2014. Proceedings, Part I*, pages 125–138, 2014.

[39] Michael W. Berry and Malu Castellanos, editors. *Survey of Text Mining II - Clustering, Classification, and Retrieval*. Springer London, 2008.

[40] Rada Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 20. Association for Computational Linguistics, 2004.

[41] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[42] Laura Plaza Morales, Alberto Díaz Esteban, and Pablo Gervás. Concept-graph based biomedical automatic summarization using ontologies. In *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*, pages 53–56. Association for Computational Linguistics, 2008.

[43] Tatsunori Mori. Information gain ratio as term weight: the case of summarization of ir results. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[44] Constantin Orăsan. Comparative evaluation of term-weighting methods for automatic summarization*. *Journal of Quantitative Linguistics*, 16(1):67–95, 2009.

[45] Constantin Orasan, Viktor Pekar, and Laura Hasler. A comparison of summarisation methods based on term specificity estimation. In *LREC*, 2004.

[46] Karolina Owczarzak. Depeval (summ): dependency-based evaluation for automatic summaries. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 190–198. Association for Computational Linguistics, 2009.

[47] Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[48] Sun Park, Dong Un An, and Youn Jeong Cho. Generic multi-document summarization using cluster refinement and nmf. In *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*, pages 65–70. IEEE, 2009.

[49] Rebecca J Passonneau. Formal and functional assessment of the pyramid method for summary content evaluation. *Natural Language Engineering*, 16(02):107–131, 2010.

[50] Dragomir R Radev and Daniel Tam. Summarization evaluation using relative utility. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 508–511. ACM, 2003.

[51] Frank Schilder and Ravikumar Kondadadi. Fastsum: fast and accurate query-based multi-document summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 205–208. Association for Computational Linguistics, 2008.

[52] Judith D Schlesinger, Mary Ellen Okurowski, John M Conroy, Dianne P O'Leary, Anthony Taylor, Jean Hobbs, and Harold T Wilson. Understanding machine performance in the context of human performance for multi-document summarization. 2002.

[53] Ehsan Shareghi and Leila Sharif Hassanabadi. Text summarization with harmony search algorithm-based sentence extraction. In *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology*, CSTST '08, pages 226–231, New York, NY, USA, 2008. ACM.

[54] S Sharma and S Rai. Genetic k-means algorithm implementation and analysis. *International Journal of Recent Technology and Engineering*, 1(2):117–120, 2012.

[55] Wei Song, Lim Cheon Choi, Soon Cheol Park, and Xiao Feng Ding. Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. *Expert Syst. Appl.*, 38(8):9112–9121, August 2011.

[56] Josef Steinberger and Karel Jezek. Text summarization: An old challenge and new approaches. In Ajith Abraham, Aboul-Ella Hassanien, André Ponce Leon F. de Carvalho, and Václav Snásel, editors, *Foundations of Computational, IntelligenceVolume 6*, volume 206 of *Studies in Computational Intelligence*, pages 127–149. Springer Berlin Heidelberg, 2009.

[57] Krysta Marie Svore, Lucy Vanderwende, and Christopher JC Burges. Enhancing single-document summarization by combining ranknet and third-party sources. In *EMNLP-CoNLL*, pages 448–457, 2007.

[58] Simone Teufel and Hans Van Halteren. Evaluating information content by factoid analysis: Human annotation and stability. In *EMNLP*, pages 419–426, 2004.

[59] Dmitry Tsarev, Mikhail Petrovskiy, and Igor Mashechkin. Using nmf-based text summarization to improve supervised and unsupervised classification. In *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*, pages 185–189. IEEE, 2011.

[60] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

[61] David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–147, 2000.

[62] Vishal Gupta and Gurpreet S. Lehal. A Survey of Text Mining Techniques and Applications. *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE*, August 2009.

[63] Xiaojun Wan. Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1137–1145, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[64] Xiaojun Wan and Jianguo Xiao. *Towards a unified approach based on affinity graph to various multi-document summarizations*. Springer, 2007.

[65] Kam-Fai Wong, Mingli Wu, and Wenjie Li. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics, 2008.

[66] Rui Xu and II Wunsch, D. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, May 2005.

[67] Chin yew Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.

[68] Pei-Ying Zhang and Cun-He Li. Automatic text summarization based on sentences clustering and extraction. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 167–170. IEEE, 2009.

[69] Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 447–454. Association for Computational Linguistics, 2006.