GReAT Model: A Model for the Automatic Generation of Semantic Relations between Text Summaries based on the Relevant Information to the User.

Claudia Gomez P. and Alexandra Pomares Q.

Pontificia Universidad Javeriana, Carrera 7 No. 40 - 62 Colombia

Abstract.

The large available amount of non-structured texts that be- long to different domains such as healthcare (e.g. medical records), justice (e.g. laws, declarations), insurance (e.g. declarations), etc. increases the effort required for the analysis of information in a decision making pro- cess. Different projects and tools have proposed strategies to reduce this complexity by classifying, summarizing or annotating the texts. Partic- ularly, text summary strategies have proven to be very useful to provide a compact view of an original text. However, the available strategies to generate these summaries do not fit very well within the domains that require take into consideration the temporal dimension of the text (e.g. a recent piece of text in a medical record is more important than a pre- vious one) and the profile of the person who requires the summary (e.g the medical specialization). To cope with these limitations this paper presents ”GReAT” a model for automatic summary generation that re- lies on natural language processing and text mining techniques to extract the most relevant information from narrative texts and discover new in- formation from the detection of related information. GReAT Model was implemented on software to be validated in a health institution where it has shown to be very useful to display a preview of the information about medical health records and discover new facts and hypotheses within the information. Several tests were executed such as Functional- ity, Usability and Performance regarding to the implemented software. In addition, precision and recall measures were applied on the results ob- tained through the implemented tool, as well as on the loss of information obtained by providing a text more shorter than the original.

Keywords:

Topic Identification, User’s Profile, Text Mining, Automatic Summary Generation Techniques, Semantic Relations

This work was supported by the project “Extraccion semi-autom´atica de metadatos de fuentes de datos estructuradas: Una aproximacion basada en agentes y miner´ıa de datos” funded by Banco Santander S.A. and Pontificia Universidad Javeriana.

DOI : 10.5121/ijcsity.2013.1405

1.Introduction

During the last thirty years the information systems have stored huge amounts of information in different formats, in some areas or domains such as healthcare (e.g. medical records), justice (e.g. sworn declarations), assurance (e.g. declara- tion) and insurance (e.g. Research articles and reports). A lot of this information is stored as narrative texts, hindering its use for the decision making processes. The process of discovering the knowledge contained in these texts, or creating new hypotheses according to them include a lot of time and effort [14],[24] that cannot be afforded by most organizations.

Generally, this problem remains a constant challenge due to the difficulty of organizations to absorb and use the information they need. It should be noted that, the limits on the reading speed of a human being make it impossible to capture the key information in a short time when there is a large amount of text [18]. Therefore, organizations need an increasingly intelligent use of information and technology to achieve a more efficient management of data they accumulate [7].

Given that the information overload on an issue creates difficulties to un- derstand it, and also, as Feldman mentioned [33], you can find the previously unknown information not only contained in a single text but in a collection of texts, or within related information for example, in electronic medical records you can find a lot of information that is contained in the annotations made by doctors such as links of interest or comparable groups exposed to a risk factor, being very useful in medical research or administrative analysis.

To achieve the proper management of this unstructured information (text) the typical approach is to use information retrieval techniques as search engines do. However, the amount of information obtained is still above what a person can handle and manage [36]. A more efficient process for the discovery of new information or knowledge known as Text Mining, allows the automatic extraction of different collections of documents for the discovery of patterns or trends that are not known [15].

To cope these problems this paper presents GReAT a model for automatic generation of semantic relations between text summaries based on the identi- fication of relevant topics that relies on natural language processing and text mining techniques to extract the most relevant information from narrative texts focused on the requirements and profile of the end user. GReAT is an extraction based model of summary generation which principle is to identify the relevant categories for the user through the text and to identify the information that is relevant to his profile generating an adequate, concise and high quality text summary. In addition, it is a model that generates semantic relations between these text summaries to provide a comprehensive overview of existing and re- lated information. Experimental results show the relevance and applicability on the tasks of the users.

This paper is organized as follows: Section 2 compares GReAT with impor- tant proposals and illustrates the main strategies of text summary generation and generation of semantic relations, Section 3 shows the General Process of the Proposed Model -

GReAT for the automatic generation of semantic rela- tions between text summaries, Section 4 describes the prototype implemented to validate the proposed model, Section 5 evaluates the functionality of GReAT through its application in a study case in the healthcare domain and finally, Section 6 presents the Conclusions and Future Work.

2 State of the Art

2.1 Generation of Summaries

To face these narrative information overload problems, different summary gen- eration approaches have been proposed. There are mainly two approaches to perform this task: Statistical and Linguistic Methods. The statistical methods are independent of the language and are based on the frequency of words or take into account the title, headings, position and the length of the sentence. On the other hand, linguistic methods include discourse structure and lexical chains [41]. Some proposals of this method use Clustering based strategies that group together phrases with similar characteristics, however, they have had accuracy problems due to the ambiguous terms of a language.

Some commercial tools are based on basic statistical approaches, and rely heavily on a particular format or writing style, such as the position of the text or some lexical words[30]. Similarly, most of the tools rely on methods or fea- tures such as centroid-based, position-based, frecuency-based summarization or keywords to extract relevant sentences to the summary such as MEAD, Dragon ToolKit [43] , LexRank [11]. There are methods based on abstraction, which are more complicated compared to approaches based on extraction, so there are still problems regarding semantic representation, inference and natural language gen- eration [42]. ML (Machine Learning) and IR (Information Retrieval) algorithms are needed to achieve a good similarity measure between documents, since there are many problems while computing the similarity between documents due to ambiguities in the used vocabulary. That is, if two documents or publications dis- cuss the same topic, they may use different vocabulary while being semantically similar.

Some aspects have been obtained through the analysis of various works that have used the most common techniques within the area of text mining for text summary generation. This analysis is consolidated in Table 1, which shows com- parative features related to how to select the relevant phrases and build up the summary, including: Frequency, Topics, Profile, Sequentiality, Duplicity and Noise. The symbol" X" indicates that the project contains the features represented in the table and the meaning of each one of these are described be- low: i) Frequency: Indicates if the project takes into account the frequency of words to extract relevant sentences to the text summary. ii) Topics: Indicates if the project takes into account the topics in the text that are important to the user to generate the text summary. iii) User Profile: Indicates if the project gives importance to the characteristics of the end-user groups to generate the text summary. iv) Sequentiality: Indicates if the project maintains the se- quentiality of the original text to form the summary in the chronological order in which it was stored. v) Duplicity: Indicates if the project eliminates the du- plicity of information in the text. vi) Noise: Indicates if the project eliminates the noise of natural language by bad writing or bad typing in the text.

Table 1. Relevance of Sentences

Project	Frequency	Topics	Profile	Sequentiality	Duplicity	Noise
[20]		X	X			
[40]	X	X	X			X
[12]	X	X				
[36]	X					
[21]	X					
[5]	X				X	
[22]	X					
[19]	X	X				
[9]		X				
[25]	X	X	X	X	X	
[24]	X					X
[32]	X					
[30]	X	X				
[11]	X					
[10]		X		X	X	
[16]	X					
[41]	X	X				X
[2]	X			X	X	

In conclusion, none of the compared projects contains all of the mentioned features and do have some limitations to be taken into account in this project with the proposed model. Among the challenges and opportunities identified in the literature to be considered for the automatic generation of text summaries are: i) To improve the quality and coherence of the summaries removing the duplicity of information, ii) Adapt summaries to every user according to their needs of information taking into account the different topics that are often im- portant to the end-user in a specific domain and user's profile. iii) Keep the sequentiality of the original document. iv) Detect noise in text collections due to the use of natural language.

2.2 Generation of Relations

In the analysis performed on the methods for the generation of relations between terms we found in most of these works the issue know as ”Problem vocabulary”, since different languages are ambiguous, the knowledge of the semantic relations between all possible words and phrases is required. To accomplish this task it is believed that one must first solve all the other problems related to Natural Language Processing such as the natural language understanding, common-sense reasoning and logical thinking, however, taking into account some semantic re- lations to relate the generated text summaries.

In the table 2 the techniques, characteristics and strategies used in the con- sulted works have been consolidated, showing some opportunities and challenges to be addressed by our proposed model for the detection of semantic relations :

Table 2. Generation of Relations

Project	Taxonomy	Profile	Matrices	Frequency	Granularity	Keywords
[25]				X		X
[28]				X		
[17]		X				
[34]			X	X		
[27]	X				X	X
[26]	X					
[14]						X
[1]				X	X	
[31]					X	

This table contains the following information:Taxonomy, Profile, Ma- trices, Frequency, Granularity and Keywords. The symbol” X” indicates that the project contains the features represented in the table and the meaning of each one of these is described below: i) Taxonomy: Indicates if the project used external knowledge within your solution or not. ii) User Profile: Indicates if the project takes or not into account the information needs or characteristics of the user. iii) Matrices: Indicates if the project stores the collection of con- cepts that are closely related semantically in a matrix of similarity, inference or interest. v) Frequency: Indicates if the project performs an analysis of the frequency of words to identify the keywords that relate the content. vi) Gran- ularity: Indicates if the project takes or not into account or not (besides the semantic similarity between the contents) one or more degrees of similarity be- tween the contents. vii) Keywords: Indicates if the project represents or not documents with keywords to estimate similarity efficiently. In conclusion neither project contains all the features mentioned, which are taken into account in our project to improve the detection and visualization of relations with the proposed model. This includes the following challenges and opportunities: i) To provide various types of relation displays allowing to cover different information needs of users, ii) Show the relations according to the needs and information granularities of the user, taking into account the different topics that are often important to a person in a specific domain, iii) Display the contents which do not have any relation, in order to find particular cases within the content, iv) Detect the high or low degree of similarity between summaries, considering the most relevant and similar words that the summaries possess.

3 GReAT Model

This model is divided into two components, the first one is responsible for ana- lyzing unstructured content found in the narrative text, allowing the realization of a summary that consolidates the most relevant information using text mining techniques and the second one is responsible for generating the semantic relations between these text summaries generated with the first component, according to the categories or topics of interest prior predefined by the user. This model is named GReAT (acronym Model for Automatic Generation of Relations be- tween Text Summaries). This model consists of a series of processes, initially in order to generate the narrative summaries, and subsequently to generate the semantic relations between text summaries generated with the first component of the model. Each one of the components is described in detail:

3.1 Component for Text Summaries Generation

Among the main concepts to keep in mind to understand the first component:

Summary: Its task is to extract a smaller document in size but keeping the relevant information from the original document, i.e., automatically creates a comprehensible version of a given text, providing useful information to the user [16]. GReAT is a model that produces summaries with the following character- istics: Multi-Document, Specific Domain, Topic Oriented and Summary Based on Extraction. It uses several techniques of Text Mining, such as: Tokenization, Chunking, Named Entity Extraction, among others.

Multi-Document: The summary is generated from multiple input docu- ments [42].
Specific Domain: The summary is generated from multiple input docu- ments, but considering the context or domain in which it was written.

Summary Based on Extraction: The summary is generated with phrases that are included literally. This strategy produces a summary by selecting a subset of sentences from the original document.

Topic Oriented approaches: It focuses on a user's topic of interest, ex- tracting text information that is related to the specific topic. Its approach is to identify significant topics within the data set and generate the topical structure based on these topics [41].
Its principles are based on taking into account some considerations to extract knowledge or generate an adequate summary such as: quality, consistency, adap- tation, duplicity, user profile, noise and sequentiality, obtained from the analysis of the state of the art detailed in the section 2.

The processes that are part of the component are presented in Figure 1. It consists of three main processes: Preprocessing, Identification of Categories and Extraction of Candidate Phrases. Preprocessing step is divided into several tasks: Stop-words, Tokenization, Spelling and Filtering, in the following sections each one of these steps is explained in detail.

At the beginning of the process, User Profile, Search Filter and Cor- pus(collection of documents) are received as input information. For instance in a health domain, the User Profile is comprised of: The keywords that identify the user's profile, in the case of a physician anesthesiologist, the keywords could be anesthesia, local, general. The Search Filter is comprised of: a date range (start date and end date) of the patients' medical records, the categories or topics of interest on which the user wants to generate the summary (including: Diseases, Drugs, Exams). The keywords associated to a category allow to specify key words that the user wants to locate, the number of sentences required be displayed from each category and a Corpus or collection of documents that will consist on the patients' medical records in text narrative.

1. Preprocessing: This step is one of the most important in the text mining area, since its results will affect the performance and quality of the data in the subsequent phases, given the large number of words, phrases and sentences there might be a large number of different forms for each of these elements to have in different contexts and combinations, therefore, this process will be a transfor- mation of the input documents, in order to extract the most significant terms or ”features” from the text. This process is divided into five main tasks: Stop- words, Tokenization, Spelling and Filtering, which are detailed below:

Stop-words: To address the problem of high dimensionality that commonly occurs in a text mining process, it is proposed to delete the words with little relevance to the language with the technique known as Stop-Words [6]. They are words that do not provide relevant information, such as articles, prepositions, among others. This process requires a dictionary of stops words of the language. High dimensionality means that the size and scale of the possible combinations of the values of the characteristics of data are large in text mining systems. The result of this phase will be the phrases of the text without the words considered as Stop-Words.

Tokenization: In order to obtain a text that is shorter than the original for the summary, it is necessary to divide the text into phrases that allow us to process information independently [13]. The output of this phase is the set of sentences that were divided by a tokenizer, in the case study the selected token were the tab spaces between lines. In addition to this, to get the information as uniform as possible, the Case folding technique is used, which seeks to convert all characters into the same kind of letter may be uppercase or lowercase letters, in the case study they becomes uppercase [38].
Spelling: Normally, natural language can make noise, ie, misspelled words. This phase performs a spelling check of the narrative text, which involves taking the text input and providing a corrected text. Spell Checking is performed on the Noisy-channel model, which models user errors (typographical) and expected user input (based on data) [8]. The errors are modeled by the weights of the Edit Distance technique and the expected input by the model language characters.

Edit distance technique measures the minimum number of edit operations (insertion, deletion, and substitution) to transform one string into another. Edit distance is selected because it can effectively capture typographic errors, words with alternative spellings, and does not rely on the separation of word boundaries [39]. This process requires a dictionary of words from the language and from specific domain. The result of this phase will be the correctly spelled phrases.

Filtering: This process proceeds to remove the redundancy of information found in these texts. To achieve this, we use a technique called Fixed Weight Edit Distance [23], where the simplest form of weighted edit distance simply sets a constant cost for each one of the edit operations: match, substitute, insert, delete, transpose. This general setup subsumes the Needlman-Wunsch algorithm used in molecular biology. This algorithm will maximize the number of matches between the sequences along the entire length of sequences. The output of this phase is the removal of common phrases to each other, leaving only one instance of them. This step is performed to eliminate duplication of information presented in the specific domain of the case of study which validated the model GReAT, therefore, it is an optional step.

2. Identification of Categories: This process consists of two tasks, Verify
Search Filter and Phrases Annotation, which are described below:

Verify Search Filter: At this point of the process the words of specific domain or Knowlegde Base, the User Profile and Search Filter have been defined. These are temporarily stored in the knowledge base to be taken into account in the next task. The output of this phase is knowlegde base with words that are relevant to the user.

Phrases Annotation: Users can have different information search needs, so in this process we will identify the most relevant information to the user accord- ing to the words associated with the topics of interest defined in the knowledge base, user profile and search filter defined. For this, it performs a process of an- notation of the words contained in the sentences resulting from the preprocessing phase.

An annotation is a layer attached to the text representation, relieving the linguistic structure in the text which can be coded as such (NN - noun), (IN- preposition) [3]. The annotations are used to identify important areas of a text that are useful to generate a summary [32].

To assign the annotations of words containing the phrases in a document to summarize, there is a technique known as Named Entity Extraction, which involves supervised training of a statistical model, or more direct methods such as a dictionary or regular expressions to classify texts or phrases of a document in a category. However,due to the lack of training data to perform the process of finding the entities or categories in the text, we will use a dictionary of specific domain and the technique used will be Chunking based on Dictionary, which aims to find adjacent words in a sentence that make sense being together. One example is ”diabetes mellitus type I”.

This phase will be based on the implementation of the matching text strings Aho-Corasick algorithm, which consists in finding all the alternatives against a dictionary independently of the number of matches or the size of the dictionary. This algorithm was invented by Alfred V. Aho and Margaret J.Corasick, being a kind of dictionary-search algorithm that locates elements of a finite set of strings (dictionary) in a text input [37].

The output of this phase is each phrase, together with the set of annotated words of the corresponding category, applying heuristics to eliminate phrases that have no set of annotated words on the categories selected by the user.

3. Extraction of Candidate Phrases: This phase involves finding key phrases to form the summary. After the preprocessing phase, which purpose was to de- bug the information and the phase of identification of phrases were performed, a list of phrases with words annotated to the categories selected by the user is generated to find the most relevant and adequate sentences. If the results are higher than the expected, it may require further filtering of information depend- ing on the size of the summary that the user has selected. For this, the selection of phrases is performed by three filters: Weighting by Categories, Weighting by Frequency and Filtering by Categories, Frequency and Sequencing :

Weighting by Categories: To find the most relevant information in the texts, the technique known as LDA (Latent Dirichlet Allocation) will be used to automatically discover relevant topics in the phrases. LDA represents documents as mixtures of topics containing words with certain probabilities. It calculates the probability that the phrase belongs to the topic taking into account the total number of words that belong to a category or topic [4].

Weighting by Frequency: To find the most important information in the texts, we look the more particular information with particular words in texts. For each one of the words that were annotated for each sentence, we calculate the ”TF-IDF” frequency on the complete document.

TF-IDF is a well known statistic measure used to evaluate how important a word is to a document corpus. The importance rises proportionally to the number of times a word appears in the document but it is countered by the frequency of the word in the corpus [12].

Filtering by Categories, Frequency and Sequencing: After the process computes the probability of each sentence into a category, the inverse frequency of the frequent words in each sentence, the total of annotated words for each phrase and the number of categories to which the phrase might belong, if the maximum number of phrases requested by the user is greater than the number of annotated sentences obtained in the ”Identification of Categories” phase, it will get the ranking of each sentence, using the previously calculated values (proba- bility, the inverse frequency, the number of annotated words and the number of categories).

This ranking is used to select the best sentences from each category. It is important to mention the application of several heuristics: 1) Only allowed to select a phrase once, even if it belongs to more than one category, avoiding duplication of information, 2) When there is a conflict between the values of the ranking of two phrases, it selects the newest phrase and 3) Annotated words that are repeated in several sentences are avoided to disallow duplication of phrases. The result of this phase is the summary of text phrases selected as the most important ones out of the original text. See Algorithm 1.

3.2 Component for Generating Relations between Text Summaries

The second part of Great Model corresponds to the detection of semantic rela- tions between summaries generated with the first component described in the previous section. This component is based on the following principles: flexibility, adaptation, singularity and granularity which are explained below:

Algorithm 1 Extraction of Candidate Phrases

```
Require: D: It is a text document, f: It's a phrase or paragraph that belongs to a document D.
Require: p: It is a word that belongs to a sentence or paragraph, a: It is a word annotated to a category, $P_f c$: Value of the
probability that a sentence f belongs to a category c, applying the " LDA technique".
Require: c: It is a category of information, U: It is the identifier of a user.
Require: $U(c) \leftarrow c \in U$: Set of categories selected by the user U, i.e.; $U(c) = \{c_1, ....c_n\}$.
Require: $f(p) \leftarrow p \in p$: Set of annotated a words p belonging to a phrase f, where $f(p) = \{p_1, ....p_n\}$.
Require: a(fr): Set of values annotated word-frequency a.
Require: $f(a) \leftarrow a \in f$: Set phrases f annotated words a belonging to the categories selected by the user U(c),
where $f(a) = \{a_1, ....a_n\}$.
while $f \exists f(a)$ {While there are phrases f with annotated words a belonging to the categories selected by the user
U(c)}. do
          while $p \exists f(p)$ {While there are p words a annotated belonging to a phrase f.} do
if a 3 a(fr) {If the word annotated a not exist in the set word-frequency values a(fr)} then
$fr = 1$ Assign frequency fr of the word annotated a a 1.
Create a value word-frequency fr the set word-frequency values p(fr).
else
$fr = fr + 1$ Adding 1 to the frequency of the word annotated a. Update frequency-word value fr
the set word-frequency values p(fr).
   end if end while
$P_f c = LDA(f(p))$ Get probability l of the set of annotated words f(p) of the phrase. Add probability $P_f c$ by
category of each sentence f.
end while
```

i) Flexibility: Allows various types of visualization of relations, such as Graph and Tree, giving the possibility to adapt the visualization to the user’s informa- tion needs, for example in the form of graph is significant for researchers where not known what information is related and what degree of similarity exists. As well as the form of tree is significant for performing administrative tasks where required to group related information.

ii) Adaptation: Allows to adapt relationships to each user according to their needs and granularities of information, taking into account the different topics that are often important for a person in a specific domain. For example it as- sociates the words that correspond to the categories of information selected by the user,

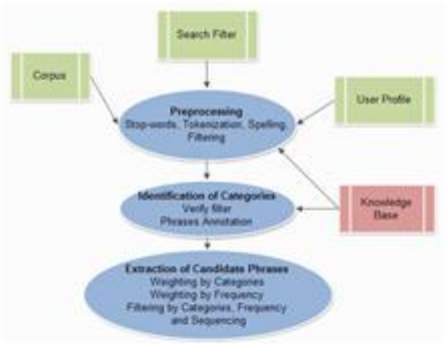
iii) Singularity: Displays content that does not have any relations, in order to find particular cases within the content,

iv) Granularity: Display varying degrees of similarity between summaries con- sidering the annotated words that they contain, the more annotated words they have in common the higher their degree of similarity.

To begin the analysis it has to be defined what a Semantic Relation is: Semantic relations between words refers to the relations of meaning between these, there are different types of semantic relations between words but for this model the semantic relations that are taken into account are the relations of Hyperonymy, Hyponymy, Meronymy and Holonym.

* Hyperonymy and hyponymy: Relation " is-a". The most general term is the hypernym for example, Flower, and the most specific one is the hyponym for example a Rose.

* Meronymy and Holonym: Establishes a hierarchy " part-of ". The set is the holonym for example Boat, and the part is the meronym for example Anchor.



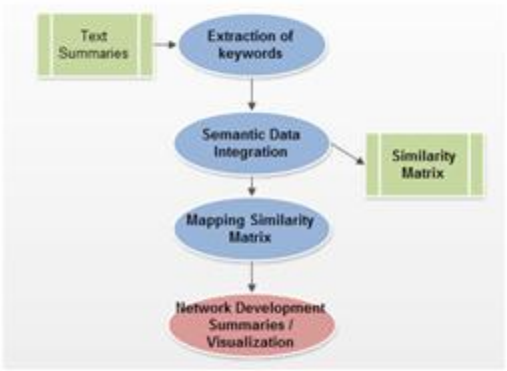

Text Summaries
Extraction of keywords
Semantic Data Integration
Similarity Matrix
Mapping Similarity Matrix
Network Development Summaries / Visualization

Fig. 1. Component for GenerationText Summarie Fig. 2. Component for Generation Rela-tions between Text Summaries

In our model the hyponym is a word that is included within the hyper- nym, ie Disease (hypernym) and Diabetes (hyponym) and a document identifier (Holonym) and the word Diabetes (Meronym).

The first two relations will be reflected in the visualization in the form of Graph, where each relation is an edge that will be identified with a color that the user has defined for each hypernym or category, and each node represented by the identifier of a document will be the hyponym. The two following relations will be reflected in the visualization in the form of tree, where each relationship will be a branch of the tree, and each parent node is represented by a word meronym and each child node of the tree represented by an identifier of a document will be the holonym. Figures 3 and 4 show an example of each of the types of visualization of relations.

The processes comprising this component are presented in Figure 2, which were obtained after analysis to consolidate the processes consulted in literature [1], [28],[34]. As a result the obtained processes are: Extraction of Keywords, Semantic Data Integration, Mapping Matrix Similarity to Data Structures and Network Development Summaries / Visualization. Each one of the component processes are described in detail:

1. Extraction of Keywords: At this stage we identify keywords in text sum- maries generated by the first component of the GReAT Model, which will be associated through a similarity measure to reflect the semantic relations be- tween the different information contents. In this process, the annotated words of each phrase of the text summaries are grouped in a vector for each summary.

2. Semantic Data Integration: During this phase a conceptual representa- tion of the data and its relations is performed. Based on the vector of keywords obtained in the previous step, a similarity matrix is constructed to generate a similarity network to be displayed. To construct this matrix, relations between summaries are quantified in terms of distance or similarity between their vector of annotated words. To calculate the similarity, we use the technique mentioned above Edit-Distance. In the analysis of keywords, identifying the relations be- tween them takes into account that the presence of a pattern or similarity implies the presence of a relation [38].

These relations will be stored in a matrix of similarity [34], where each di- mension contains the identifier of the document that was summarized and within each matrix value, the common keywords between two pairs of identifier text summaries of a document are stored. The end result of this phase will be the similarity matrix formed between the identifiers of the documents that were summarized in the first component of the model.

3. Mapping Matrix Similarity: In the previous process we obtained a sim- ilarity matrix where the relations obtained after the process of comparison be- tween the keywords of each summary are stored, using (dot product - Cartesian product) [25], between each set of words and each summary. This matrix must be mapped to a data structure that allows to develop more easily the network between text summaries. For this mapping, the comparisons between the values stored in the matrix are used to record properly in a data structure the nodes and edges of the network. The end result of this phase are two data structures that will allow displaying the two forms of visualization that the Model GReAT offers Graph and Tree.

4. Network Development Summaries / Visualization (Graph and Tree): For the process in which the network graphic summaries are developed, we found in the literature several points that are taken into account for this process: [29],[38]. On consolidation, the tasks that follow this model for network sum- maries are:

1) Collect data to visualize. 2) Analysis and mapping of the data to be visual- ized. 3) Deployment of the visualization interface.

In detail, at this process a feature extraction and key words indexing are used to construct a graphical representation of the collection of documents and to enable the user to quickly identify the main topics or concepts by its importance in rendering. With the graphical representation as a graph and tree, it is easy to discover the location of specific documents and to explore the semantic relations that exist in a large collection of documents and representing semantic features in order to visualize their meaning. The types of visualization and a description of their meaning are described below:

1) Type of Visualization - Tree: In the tree view Holonym and Meronymy relations between words annotated from the categories the user wants to see, are depicted, where the holonym is an identifier of a document or medical record according to the case study (see Figure 5) such HC1715059, and the meronym is a word that is part of the document or medical record, such as WARFARIN (see Figure 4). This type of visualization was defined to address the needs of users' groups as administrative, those requiring to identify comparable groups. For example, detecting comparable groups exposed to risk factors.

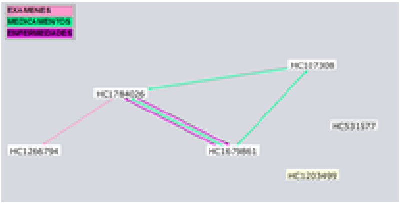


Fig. 3. Graph Visualization - Tree

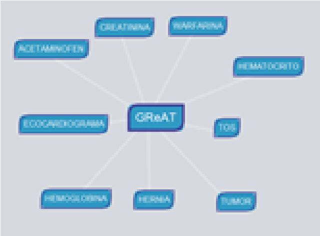


Fig. 4. Visualization - Tree

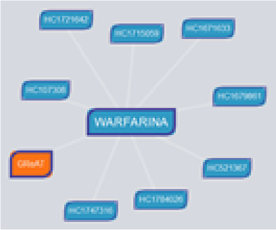


Fig. 5. Detail of the

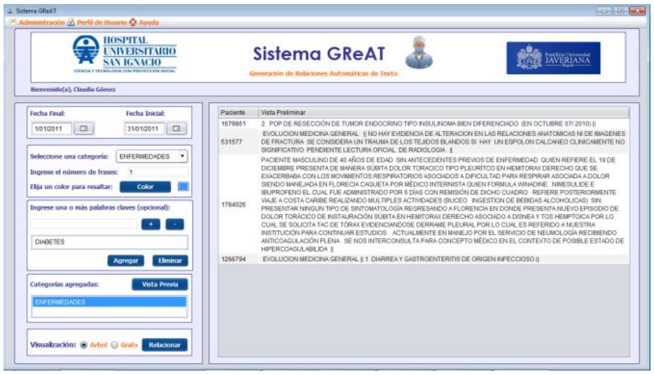
2) Type of Visualization - Graph: The graph view in the Figure 3 de- picts Hyperonymy and Hyponymy relations between words annotated from the categories that the user desires to consult, the edge of the graph represents the category which is the hypernym, for example, MEDICINES, and the hyponym is the node representing the word that is associated to the category such as WARFARIN. This type of visualization was defined to meet the needs of users' groups as researchers, who require to search and to analyze related informa- tion beforehand unknown and the degree of similarity or disparity between this information. For example, links of interest or detecting rare cases.

4 Prototype

To validate the proposed Model GReAT, it was implemented on a software called ”Great System” . This software was applied to a case study to summarize information from unstructured electronic medical records in a Colombian Hospital. This chapter initially presents how the GReAT System was implemented and later, its application in the case study. The prototype was implemented in Java, using the libraries for Natural Language Processing Lingpipe and JOrtho, as a result of the analysis of the tools and techniques for the GReAT Model like the most appropriate and complete tools to implement GReAT Model. The architecture was defined using different views, models and diagrams to document and support the decisions taken during the architectural definition of the system. The architecture is multi-layered, with three main layers: Presentation, Business and Persistence. It uses the frameworks iBatis for data access in the Persistence layer, Spring for business logic in the Business layer and Swing for the presentation layer. The main screen of GReAT System can be seen in Figure

6, along with its filter options and menu options:

For the case study there were implemented these filter options: initial date, end date, categories, total of phrases by category, color, keywords and the types of visualization (Tree and Graph). The prototype implementation allows the performing by the user of the following functions: 1. Store categories of required information. 2. Store the knowledge base of the required categories.


Sistema GReAT
Administración
Perfil de Usuario
Ayuda
HOSPITAL UNIVERSITARIO SAN IGNACIO
CIENCIA Y TECNOLOGÍA CON PROYECCIÓN SOCIAL
Sistema GReAT
Generación de Relaciones Automáticas de Texto
Pontificia Universidad JAVERIANA
Color
+
-
Agregar
Eliminar
Vista Previa
Relacionar

Bienvenido(a), Claudia Gómez

Fecha Final: 1/01/2011

Fecha Inicial: 31/01/2011

Seleccione una categoría: ENFERMEDADES

Ingrese el número de frases: 1

Elija un color para resaltar:

Ingrese una o más palabras claves (opcional): DIABETES

Categorías agregadas: ENFERMEDADES

Visualización: ● Arbol ○ Grafo

Paciente	Vista Preliminar
1679861	2 POP DE RESECCIÓN DE TUMOR ENDOCRINO TIPO INSULINOMA BIEN DIFERENCIADO (EN OCTUBRE 07/ 2010) ||
531577	EVOLUCION MEDICINA GENERAL || NO HAY EVIDENCIA DE ALTERACION EN LAS RELACIONES ANATOMICAS NI DE IMAGENES DE FRACTURA SE CONSIDERA UN TRAUMA DE LOS TEJIDOS BLANDOS SI HAY UN ESPOLON CALCANEO CLINICAMENTE NO SIGNIFICATIVO PENDIENTE LECTURA OFICIAL DE RADIOLOGIA ||
1784026	PACIENTE MASCULINO DE 40 AÑOS DE EDAD SIN ANTECEDENTES PREVIOS DE ENFERMEDAD QUIEN REFIERE EL 19 DE DICIEMBRE PRESENTA DE MANERA SÚBITA DOLOR TORACICO TIPO PLEURÍTCO EN HEMITORAX DERECHO QUE SE EXACERBABA CON LOS MOVIMIENTOS RESPIRATORIOS ASOCIADOS A DIFICULTAD PARA RESPIRAR ASOCIADA A DOLOR SIENDO MANEJADA EN FLORECIA CAQUETA POR MÉDICO INTERNISTA QUIEN FORMULA WINADINE NIMESULIDE E IBUPROFENO EL CUAL FUE ADMINISTRADO POR 6 DÍAS CON REMISIÓN DE DICHO CUADRO REFIERE POSTERIORMENTE VIAJE A COSTA CARIBE REALIZANDO MULTIPLES ACTIVIDADES (BUCEO INGESTION DE BEBIDAS ALCOHOLICAS) SIN PRESENTAR NINGUN TIPO DE SINTOMATOLOGÍA REGRESANDO A FLORENCIA EN DONDE PRESENTA NUEVO EPISODIO DE DOLOR TORÁCICO DE INSTAURACIÓN SÚBITA EN HEMITORAX DERECHO ASOCIADO A DISNEA Y TOS HEMPTOICA POR LO CUAL SE SOLICITA TAC DE TÓRAX EVIDENCIANDOSE DERRAME PLEURAL POR LO CUAL ES REFERIDO A NUESTRA INSTITUCIÓN PARA CONTINUAR ESTUDIOS ACTUALMENTE EN MANEJO POR EL SERVICIO DE NEUMOLOGÍA RECIBIENDO ANTICOAGULACIÓN PLENA SE NOS INTERCONSULTA PARA CONCEPTO MÉDICO EN EL CONTEXTO DE POSIBLE ESTADO DE HIPERCOAGULABILIDA ||
1266794	EVOLUCION MEDICINA GENERAL || 1 DIARREA Y GASTROENTERITIS DE ORIGEN INFECCIOSO ||

Fig. 6. Screens

3. Store the keywords of user’s profile. 4. Enter data filter to generate text summaries on unstructured text of electronic medical records. 5. Displaying results of generated summaries and finally, 6. Build relationships between the results obtained in the form of Graph or Tree. These options allow the user to perform a search according to his needs and profile, some of them were created considering the case study of a Hospital in Colombia.

5 Application on Case Study

The selected case study for validation of the proposed model is presented in a Hospital in Colombia, which stores information about the diagnosis, treatment and monitoring of patients through a narrative text that is found in the patient's medical records. With the case study several tests of functionality, usability and performance were conducted.

5.1 Functional tests

In order to evaluate the generated summaries, we wanted to determine how much loss of information we get with the generation of the summary. Obviously some information to generate summaries should be removed, so what was intended was to measure what relevant or irrelevant information was lost because of the summaries. What we want is that irrelevant information gets omitted, while the relevant information remains, so what we did was to compare the original document versus the summary, what we won and what was lost, which was measured in terms of the relevant phrases and sentences retrieved using precision and recall measures. To do this, we defined the relevance of a sentence taking into account the hypothesis where information retrieval systems must be measured indicating the degree of similarity between the question and the answer, in our case the question will be the search filter, the knowledge base and user profile defined by the user and the answer will be sentences in the summary, and for that you should identify any measurable property group to estimate how many of these properties are shared between the two entities. Measurable properties were defined: Does it have words associated with the knowledge base defined?, Does it have Keywords associated with search filter? or Does it have words defined in the User Profile?, what defines that a sentence is relevant or not, is that at least the phrase comply with any of these properties. In addition to this, we apply a formula to measure the loss of information. Relevance function is defined as follows:

1. Representation of sentences: a phrase is considered to be characterized as a set of one or more topics that represent its content. Following the model popularized by Salton and McGill [35], if we have a set tcp of terms or words, such that, for example, tcp = (Diseases, Drugs, Symptoms, T reatments) represents the topics of the sentences, you can generalize and transform the same expression: $Tcp = (t1, t2, t3, t4, t5)(1)$, where t1, t2, etc.., symbolize a category c or word p belonging to a sentence.

   Since this represents a phrase as a vector which adopts for example, as follows: $Fi =< 1, 1, 1, 0, 0 >$ which means that phrase i contains the words or categories t1, t2 and t3 of set tcp, where these are the words or categories that contains a phrase. Each sentence will result in a vector with a different configuration.

2. Representation of Relevant Entities: defined Relevant Entities are also represented as a vector. For example: $Er =< 0, 1, 0, 1, 1 >$, means that a particular information need has been indexed with the words or categories t2, t4 and t5, forming relevant entities Er.

Now you can see what phrases are more like the features selected by the user or profile, setting a threshold somewhere below which a sentence would be considered no longer relevant. For the case study, the established threshold was 1, which means that each sentence should at least have one relevant entity to be relevant.

3. Calculation: For example, for simplicity we have only two sentences (Fi, Fj), which present the following vectors of words or categories belonging to each phrase $Fi =< 1, 1, 1, 0, 0 >$, $Fj =< 1, 1, 0, 0, 1 >$.

   The selection of the more relevant phrase to the user is performed by cal- culating which one of the two sentences has more in common with the relevant entities. There are several ways to perform this calculation (Salton and McGill [35]). One of the simplest is the sum of products, the two numbers in each column are multiplied together, and the results are summed, as shown in Table 3. The representation of a sentence Fi is a vector of length n: $Fi =< ti1, ti2, ...tin >$ (2a), the representation of a word or category or relevant entity Er is a vector of length n: $Er =< er1, er2, ...ern >$ (2b) and the equation that allows the calcu- lation of the degree of similarity between a sentence Fi and words or categories or relevant entities Er is formalized as follows: $SIM(Fi, Er) = \mathbf{P}n \ ti1 * er1$

   $er=1$

   (3), which reads: the similarity (SIM) between the phrase Fi and Er relevant entity is equal to the sum of the products of each pair of respective vector ele- ments. For example: these are the vectors of a sentence and the relevant entities, respectively: $Fi =< 1, 1, 0, 1, 0 >$, $Er =< 0, 1, 0, 1, 0 >$, Then, the similarity is:

   $$SIM(Fi, Er) = (1 * 0) + (1 * 1) + (0 * 0) + (1 * 1) + (0 * 0) = 0 + 1 + 0 + 1 + 0 = 2.$$

Table 3. Example - two sentences

$Er = 0, 1, 0, 1, 1$	$Er = 0, 1, 0, 1, 1$
$Fi = 1, 1, 1, 0, 0$	$Fj = 1, 1, 0, 0, 1$
$0 + 1 + 0 + 0 + 0 = 1$	$0 + 1 + 0 + 0 + 1 = 2$

According to this example, the phrase Fj is the most similar to the relevant entities defined by the user. Then, to calculate the relevance of each phrase, we proceed to perform the calculations of the indices of Precision and Recall to finally get the value of information loss. The formulas of these indices are shown below:

$$Precision = \frac{\#\{phrasesrelevant \cap retrievedphrases\}}{\#phrasesretrieved},$$

$$Recall = \frac{\#\{phrasesrelevant \cap retrievedphrases\}}{\#phrasesrelevanttrue}$$

The formula for the loss of information is applied with the above results as follows: $IL = Totalrelevantphrases - Totalphrasesretrieved$

After running 14 different test cases, the average yielded a precision high value (1.0) where it is kept constant because the calculations to obtain the weighting of each sentence to select the most relevant phrases matching function proposed relevance except in some calculations. And the average yielded a high value on the completeness (0.81) being dependent on the number of phrases that the user selected to generate the summary, as this number is greater, the completeness increases handing all relevant phrases. To assess the functionality of generating semantic relations between text summaries, we successfully executed various test cases.

5.2 Usability tests

Usability tests were divided into two phases: An initial phase where System Great was introduced to a set of doctors with different specialties (cardiology, pulmonary specialist, epidemiologist, internist) and with administrative staff to validate the applicability of the model and receive preliminary comments, from where we obtained several comments. In the second phase, the comments were implemented and the same procedure was performed with internists and with the administrative director of Epidemiology and Demography in the Ministry of Health and Social Protection.

In both phases the instrument or EUCS approach was applied which was designed to measure the satisfaction of users who directly interact with a specific application called EUCS (End-User Computing Satisfaction Construct). Response options are in a range of 1-5, where the value of 1 means the user disagrees with the question, and the value of 5 means the user is more in line with the question. The results in the first phase with the mean values were not satisfactory (3.74 / 5) because various comments were expressed such as: include keywords for each category and focus the summary as a preview of the information. The results in the second phase were satisfactory (4.54 / 5).

5.3 Performance tests

In order to validate the performance of the application and describe its behavior in terms of processing speed in the process of generating text summaries several test cases were executed with historical records of the case study. The results were satisfactory as they were achieved in less processing time than DragonToolKit free tool as shown in Figure 7 reflecting the total medical records processed with both tools versus processing time resulting in milliseconds. Why compared to the tool DragonToolKit? The analysis of the knowledge base was summarized, and we found that it was the most complete tool for generating summaries, however, there were some drawbacks such as: to fulfill the maximum size defined for the summary, it trimmed phrases. Also to run the summary, you must manually cre- ate a file for each document to be summarized, so only the comparison between the two tools was made up to 65 records and Great System analysis is made up with 600 (equivalent to 1 year medical records).

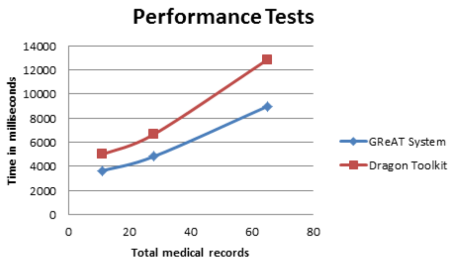


Fig. 7. Time in milliseconds versus Total medical records

6 Conclusions

From the review of the literature about the solutions that currently exist for the automated generation of text summaries and generation of semantic relations, this model seeks to address some weaknesses and challenges. The initial results show an improvement in quality and consistency of obtained summaries, taking into account the user needs and topics that are often important within the domain, the sequentiality of the original text, the duplication of information and the noise that natural language presents. On the other hand, we see the importance to relate summaries of text obtained to get the information more complete and global for the user. As future work this model can be reused in their various processes used as pre-processing process, serving as the basis for other text mining projects and the validity and usefulness of the proposed model can be extended further considering other semantic relations such as antonyms and synonyms.

References

[1]. A text-mining-based patent network: Analytical tool for high-technology trend. The Journal of High Technology Management Research, 15(1):37 – 50, 2004.

[2]. Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT ’11, pages 500–509, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[3]. D. B.Tsujii J. Ananiadou, S.Kell. Text mining and its potential applications in systems biology. Trends in Biotechnology, 2006.

[4]. Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation based multi-document summarization. In Proceedings of the second workshop on Analytics for noisy unstructured text data, AND ’08, pages 91–97, New York, NY, USA, 2008. ACM.

[5]. Aur´elien Bossard, Michel G´en´ereux, and Thierry Poibeau. Cbseas, a summarization system integration of opinion mining techniques to summarize blogs. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session, EACL ’09, pages 5–8, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[6]. Senso Jose A. Brun Ricardo Eto. Miner´ıa textual. IEEE, 2004.

[7]. Alonso Martinez Margarita Cobo Ortega Angel, Rocha Blanco Rocio. Descubrim- iento de conocimiento en repositorios documentales mediante t´ecnicas de miner´ıa de texto y swarm intelligence. In Rect@ Vol 10 Diciembre 2009.

[8]. Hal Daum´e, III and Daniel Marcu. A noisy-channel model for document compression. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02, pages 449–456, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[9]. C.L. Devasena. Automatic text categorization and summarization using rule reduction. In Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on, pages 594–598, 2012.

[10]. Pierre-Etienne Genest and Guy Lapalme. Framework for abstractive summariza- tion using text-to-text generation. In Proceedings of the Workshop on Monolingual Text-To-Text Generation, pages 64–73, Portland, Oregon, June 2011. Association for Computational Linguistics.

[11]. Dragomir R. Radev Gunen, Erkan. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research 22 (2004) 457-479, 22, 2004.

[12]. Jung-Hsien Chiang Heng-Hui Liu, Yi-Ting Huang. A study on paragraph ranking and recommendation by topic information retrieval from biomedical literature. IEEE, 2010.

[13]. Paass G. Hotho A., Nurnberger A. A brief survey of text mining. IEEE, 2005.

[14]. Tasha R. Inniss, John R. Lee, Marc Light, Michael A. Grassi, George Thomas, and Andrew B. Williams. Towards applying text mining and natural language pro- cessing for biomedical ontology acquisition. In Proceedings of the 1st international workshop on Text mining in bioinformatics, TMBIO ’06, pages 7–14, New York, NY, USA, 2006. ACM.

[15]. Theo j.d. Bothma Jan h. Kroeze, Machdel c. Matthee. Differentiating data- and text-mining terminology. ACM, 2003.

[16]. Keivan Kianmehr, Shang Gao, Jawad Attari, M. Mushfiqur Rahman, Kofi Akomeah, Reda Alhajj, Jon Rokne, and Ken Barker. Text summarization tech- niques: Svm versus neural networks. In Proceedings of the 11th International Con- ference on Information Integration and Web-based Applications y Services, iiWAS ’09, pages 487–491, New York, NY, USA, 2009. ACM.

[17]. A.Kongthon, C. Haruechaiyasak, and S. Thaiprayoon. Constructing term the- saurus using text association rule mining. In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008. 5th International Conference on, volume 1, pages 137–140, 2008.

[18]. Dr Richi Nayak Lin Chen. A case study of failure mode analysis with text mining methods. IEEE, 2007.

[19]. Chong Long, Min-Lie Huang, Xiao-Yan Zhu, and Ming Li. A new approach for multi-document update summarization. J. Comput. Sci. Technol., 25(4):739–749, July 2010.

[20]. M A Zaveri M K Dalal. Heuristics based automatic text summarization of un- structured text. ACM, 2011.

[21]. Dr. B. Ravindran M. Saravanan, Dr. S .Raman. A probabilistic approach to multi- document summarization for generating a tiled summary. IEEE, 2005.

[22]. Antnio Horta Branco Marcus V. C. Guelpeli Ana Cristina B. Garcia. The process of summarization in the pre-processing stage in order to improve measurement of texts), journal = IEEE, year = 2011.

[23]. Yashar Mehdad, Matteo Negri, Elena Cabrio, Milen Kouylekov, and Bernardo Magnini. EDITS: An Open Source Framework for Recognizing Textual Entailment.

[24]. Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukr- ishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. Using citations to generate surveys of scientific paradigms. In Proceedings of Human Language Tech- nologies: The 2009 Annual Conference of the North American Chapter of the As- sociation for Computational Linguistics, NAACL ’09, pages 584–592, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[25]. Pradeep Muthukrishnan, Dragomir Radev, and Qiaozhu Mei. Simultaneous sim- ilarity learning and feature-weight learning for document clustering. In Pro- ceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing, TextGraphs-6, pages 42–50, Stroudsburg, PA, USA, 2011. Association for Compu- tational Linguistics.

[26]. S. Nadschlager, H. Kosorus, A. Bogl, and J. Kung. Content-based recommen- dations within a qa system using the hierarchical structure of a domain-specific taxonomy. In Database and Expert Systems Applications (DEXA), 2012 23rd In- ternational Workshop on, pages 88–92, 2012.

[27]. S. Nadschlager, H. Kosorus, P. Regner, and J. Kung. Semantic data integration and relationship identification using the hierarchical structure of a domain-specific taxonomy. In Database and Expert Systems Applications (DEXA), 2012 23rd In- ternational Workshop on, pages 48–52, 2012.

[28]. Na Ni, Kai Liu, and YaoDong Li. An automatic multi-domain thesauri construction method based on lda. In Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on, volume 2, pages 235–240, 2011.

[29]. Zhou Ning, Wu Jiaxin, Wang Bing, and Zhang Shaolong. A visualization model for information resources management. In Information Visualisation, 2008. IV ’08. 12th International Conference, pages 57–62, 2008.

[30]. Fukuhara Tomohiro Park Jaehui. Web content summarization using social book marks: A new approach for social summarization. ACM, 2008.

[31]. Aurora Pons-Porrata, Rafael Berlanga-Llavori, and Jose Ruiz-Shulcloper. Topic discovery based on text mining techniques. Information Processing y Management, 43(3):752 – 768, 2007. Special Issue on Heterogeneous and Distributed IR.

[32]. Lawrence H. Reeve, Hyoil Han, Saya V. Nagori, Jonathan C. Yang, Tamara A. Schwimmer, and Ari D. Brooks. Concept frequency distribution in biomedical text summarization. In Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM ’06, pages 604–611, New York, NY, USA, 2006. ACM.

[33]. James Sanger Ronen Feldman. The text mining handbook. Cambridge University Press, 2 edition, 2007.

[34]. Zhao J.Leon Roussinov Dmitri. Automatic discovery of similarity relationships through web mining. Elsevier Sciense, 2002.

[35]. Gerard Salton and Michael J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA, 1986.

[36]. Wen-Feng Hsiao Te-Min Chang. A hybrid approach to automatic text summarization. IEEE, 2008.

[37]. Nhat-Phuong Tran, Myungho Lee, Sugwon Hong, and Minho Shin. Memory efficient parallelization for aho-corasick algorithm on a gpu. In High Performance Computing and Communication 2012 IEEE 9th International Conference on Em- bedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Con- ference on, pages 432–438, 2012.

[38]. Gurpreet S. Lehal Vishal Gupta. A survey of text mining techniques and applications. JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLI- GENCE, 2009.

[39]. Wei Wang, Chuan Xiao, Xuemin Lin, and Chengqi Zhang. Efficient approximate entity extraction with edit distance constraints. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, SIGMOD ’09, pages 759–770, New York, NY, USA, 2009. ACM.

[40]. ChengXiang Zhai Bruce Schatz Xu Ling, Qiaozhu Mei. Mining multi-faceted overviews of arbitrary topics in a text collection. ACM, 2008.

[41]. Jiaming Zhan, Han Tong Loh, and Ying Liu. Gather customer concerns from online product reviews - a text summarization approach. Expert Syst. Appl., 36(2):2107– 2115, March 2009.

[42]. LI Cun-he Zhang Pei-ying. Automatic text summarization based on sentences clustering and extraction. IEEE, 2009.

[43]. Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. Dragon toolkit: Incorporating auto-learned semantic knowledge into large-scale text retrieval and mining. In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, volume 2, pages 197–201, 2007.