Available online at www.sciencedirect.com
ScienceDirect
Data & Knowledge Engineering 64 (2008) 330–345
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries

Christos Bouras *, Vassilis Poulopoulos [1], Vassilis Tsogkas [1]

Research Academic Computer Technology Institute, N. Kazantzaki, Panepistimioupoli and Computer Engineering and Informatics Department, University of Patras, 26500 Rion, Patras, Greece

Received 29 March 2007; accepted 31 July 2007
Available online 11 September 2007

Abstract

In this manuscript we present the summarization and categorization subsystems of a complete mechanism that begins with web-page fetching and concludes with representation of the collected data to the end users through a personalized portal. The system intends to collect articles from major news portals and, following an algorithmic procedure, to create a more user friendly and personalized "view" of the articles. Before presenting the information back to the end user, the core of our mechanism automatically categorizes data and then extracts personalized summaries. We focalize to the core of the mechanism and more specifically, we present the algorithms used for the summarization and the categorization of texts. The algorithms are not utilized only for producing isolated data, targeted to a specific subsystem, but a combination of the algorithms, which achieves co-operation of the categorization and summarization mechanisms, is introduced in order to enhance text labeling through the personalized summaries that are constructed.

Keywords: Summarization algorithms; Categorization procedure; Data reprocessing; Efficient summarization; Text summarization; Personalizing the web; Text categorization; Sentence weighting

1. Introduction

We are witnessing an era that internet users have reached outrageous numbers. Additionally, the web pages along with the information that resides in them create a chaotic condition for the World Wide Web. This condition is neither static nor stable, but a dynamic, continuously changing one that feeds daily the entropy of this chaotic system. The consequence of the popularity of the web, as a global information system, is that it is flooded with a large amount of data and information, thus making the task of locating useful information a tedious and frustrating experience. Many attempts have been made in order to count the amount of existing web pages and the estimation of more that 10 billion web pages, seems to be conservative. Furthermore, each

* Corresponding author. Tel.: +30 2610 960375.
E-mail addresses: bouras@cti.gr (C. Bouras), poulop@cti.gr (V. Poulopoulos), tsogkas@cti.gr (V. Tsogkas).
[1] Tel.: +30 2610 996954.

0169-023X/$ - see front matter
doi:10.1016/j.datak.2007.07.007

of these pages vary from including, no information at all, to thousands of pages full of information, multimedia and news articles. The problem from the afore-mentioned condition arises when searching for useful information. In our work, we focus this searching on news and articles deriving from multiple news portals. From a brief search, we have located more than thirty major or minor news portals existing in America that include worldwide news (concerning probably not just US users). This means that whenever a user needs to be informed about an issue he/she has to search through all the news portals one by one. This condition is quite common nowadays for the internet users and could be considered as the problem of locating useful information among many news portals as well as the ability to track down on a specific news topic on a daily basis. The idea for constructing our mechanism derives from the fact that neither the searches within the news portals nor the unfiltered RSS feeds can provide an affordable solution for the afore-mentioned problem. The mechanism that we have constructed collects news articles from major news portals, extracts the keywords that characterize the articles, categorize the articles according to a set of representative categories, create automatically generic summaries of the news articles and finally present the articles to the Internet user through a personalized web site or through dynamically created RSS feeds. In this work we focalize on the core of our mechanism which consists of the summarization and categorization mechanisms. Our intention is to highlight how the co-operation of the summarization and categorization can lead to more efficient text labeling. More specifically the idea relies on the fact that a summary includes only important information from an article and thus more representative information concerning the category to which the article belongs. Text searching and summarization are two critical methods for resolving part of the afore-mentioned problem. The search engines play a filtering role for the information while text summarizers are utilized as information spotters to help users identify a final set of desired documents [14]. Automatic text summarization is the process of distilling the most important information from a set of sources to produce an abridged version for a particular user and task. Recently, there have been many efforts towards the direction of text summarization together with the many forms it can take, e.g. Web page summarization [18,3], online encyclopedia summarization [7], etc. In our work, we focus on the summarization mechanism, as well as, the interactions between the summarization and categorization mechanisms of our system. More specifically, we evaluate the performance of the summarization module, describe the algorithmic procedure that leads to personalization of the articles' summary based on sentence weighting, and explain the algorithmic procedure that leads to better results for each mechanism with the support of the other. Starting from a training set of documents, we generated some basic categories. Then we used a set of articles on a daily basis as input to the mechanism and applied the summarization and categorization algorithms to them. During this procedure, we tried to estimate the way the results of the summarization could affect the categorization procedure and vice versa. Additionally, we found a limit for each of the procedures that can lead to most efficient results for both mechanisms. According to the distinction of knowledge-poor and knowledge-rich categories for the summarization techniques, our approach could be characterized as knowledge-poor because the basic algorithm for summarization is based on heuristics. Though, the interaction between the categorization and summarization modules enables the summarization to obtain some kind of ''knowledge'' about the domain of the keywords. This implies that the mechanism introduces an algorithm for a new category of summaries that lies between the knowledge-poor and knowledge-rich categories. The remaining of our work is structured as follows. In the next section we present the related work concerning the research interest of our work. In Section 3, an overview of the proposed mechanism, peRSSonal, is presented. Section 4 includes the general architecture of the whole mechanism and focuses on the core of our system which is described thoroughly in Section 5. In Section 6 we present the experimental results of our work and we conclude in Section 7 with some remarks about the mechanism and future work.

2. Related work

The procedure of creating efficient, automatic text summaries begins from the late 1950s with the analytic approach from Luhn [13], whose classic work is based on analysis of words and sentences. Some techniques [6,15] introduce the seeking of special words or phrases in the text, while others are based on patterns of relationship between sentences, or take into consideration the length of the sentences [16,12]. More advanced techniques do not use elements from the ''corpus'' (the set of documents on which summarization is applied) itself,

but try to generate the text directly using a knowledge-based representation of the content, or a statistical model of the text [20,3]. Even probabilistic models of term distribution in the documents are researched in order to create summaries of corpuses [17].

The summarization techniques are roughly divided into four categories. The first category contains techniques that use some kind of heuristic approach towards the problem. Sentence rating or special weighting of sentences containing title words [6] belong to this category. The second category includes corpus-based methods [12] that frequently use the TF-IDF (term frequency – inverse document frequency) method. The third category includes methods that take into account the text structure. Lexical chains usage is a representative method of this class [2]. Finally there is a category that uses knowledge-rich approaches towards the problem. Summarization methods of this category are the most advanced but are of use only for particular domains. An effort for an online medical encyclopedia is presented in [7].

Another categorization of the summarization techniques is introduced by Mani and Hahn [8] concerning the extent of involvement of domain-knowledge. The two categories define methods that are knowledge-poor or knowledge-rich. The first category includes methods that do not take into account any knowledge-specific domain and thus can be easily applied to any domain, while knowledge-rich techniques assume that knowing or understanding the meaning of the text will lead to better results. According to this ontology, heuristics and TF-IDF are considered to be knowledge-poor, while knowledge-based and statistical models are knowledge-rich techniques.

Recently, in [11] there is an effort to find the dynamic portions of a document and use this to produce good summaries based on the hypothesis that the higher the number of dynamic parts containing a term, the more important this term is for the summary. In [18], the writers try to adopt web-page summarization to web-page classification and improve the classification results using summarization methods. Using text categorization to produce good summaries is also faced in [1] where the writers use a self-organizing feature map (SOFM) which learns the salient features of each of the texts and assigns the text into a mnemonic position of the map. Latent semantic analysis [19] is also frequently used for extracting summaries. NLP, while not always the best choice, is used frequently, e.g. the SUMMARIST system [9]. These methods tend to operate at word level and miss concept-level generalizations. Marginal Relevance (MMR) holds the idea of balancing novelty and usefulness of terms and focuses on query-based summarization of a static collection of stories. In many of the techniques, the problem is faced as a classic IR problem and solved using precision-recall metrics.

Text classification (categorization) is the process of deciding on the appropriate category for a given document. Classification tasks include determining the topic area of an essay; deciding to what folder an email message should be directed; and deciding to which newsgroup a news article belongs. The purpose of text categorization [10] is to accompany readers to their search of news articles, by creating and maintaining key categories which hold articles related with a specific topic of interest. New articles are categorized to the predefined set of categories using some criteria which vary from one categorization technique to another. The use of predefined categories can be relatively coarse-grained, i.e. only some basic, unrelated to each other, categories are defined, such as business, education, and science, or fine-grained where many categories, which are frequently overlapping with each other, are introduced.

3. The peRSSonal mechanism

The main idea for constructing the mechanism lies on the fact that the daily Internet life has changed recently with every "corner" of the WWW becoming a source of information. Thus, searching for information often becomes a tedious task. The problem of refining the search is resolved, or tried to be resolved through the major search engines (e.g. Google, Yahoo), or through attempts of creating ontologies of the WWW (e.g. DMOZ). The problem is extreme as every internet user has different special needs from the medium that is called World Wide Web. From our experience, we realized that among these problems (searching for information) lies another huge problem that concerns millions of users around the globe. As the Internet expands and acts as a form of information, more and more people realize that they are able to read and stay informed by articles from around the globe in real time. It is not coincidental that the most well-known websites after the major search engines are the news portals (list of top 100 websites). A huge problem that arises from these facts is that the users have to visit every single news portal and read the news from the categories that are

concerned. The RSS feeds and the personalized microsites resolve partially the problem. In the first occasion the users do not have to visit every single website but they still have to filter the information as the feeds are not focalized on each user's specific needs. The second solution is focalized on each users needs but still the users have to visit every single website in order to collect or track all the information about a specific new. Our mechanism intends to resolve all the afore-mentioned problems through a single, personalized website that can also offer dynamically created personalized RSS feeds. In order to achieve this, a specific procedure has to be followed. First, collect all the news articles from the major news portals in real time. Secondly, categorize the fetched articles and extract a generic summary for each article. Finally, present the refined articles to the users of the website in a personalized view without any additional information that is useless for the needs of the Internet users. In this manuscript we focalize on the core of our mechanism which is the automatic categorization subsystem and the summarizer. In order to enhance the quality of information that is presented to the end user, we realized that the core of our mechanism should produce more refined results. Thus, we had to examine the best algorithms for each of the core subsystems and decide on the best solution. Additionally, we decided to design a solution in order to interconnect the functionality of the core modules in order to extract better results. We noticed that a categorized text can provide feedback to the summarization mechanism and thus create better summaries for the articles and on the other hand, the concision of information of a summary can improve the functionality of the categorization mechanism. In this paper we present how we managed to interconnect the subsystems of the core mechanism and the algorithms that we have utilized in order to produce the desired result.

4. Architecture

The architecture of the system is distributed and based on standalone subsystems but the procedure to produce the desired result is actually sequential. This means that the flow of information is representative of the subsystems that the mechanism consists of. Another important architectural issue is the modularity of the mechanism. In this section we will describe how these features are implemented through the architecture of the system. We are putting the focus on the module of text summarization, though, analysis of the categorization module and the personalization mechanism of the personalized portal is presented in order to cross-connect the features of our system. As already mentioned, the summarization procedure acquires information from the pre-processing procedure and exchanges knowledge with the categorization and personalization mechanisms in order to format each text's summary according to each user's needs.

The mechanism consists of a series of subsystems that produce the desired result. The collaboration between the distributed systems is based on the open standards for input and output that are supported by each part of the system and by communication with a centralized database. Fig. 1 depicts the architecture of the complete mechanism.

The procedure of the mechanism, as depicted in Fig. 1, is: (a) capture pages from the www and extract the useful text, (b) parse the extracted text, (c) summarize and categorize the text, and (d) present the personalized results to the end user. In order to capture the pages, a simple crawler is used. The addresses that are used as input to the crawler are extracted from RSS feeds. The RSS feeds point directly to pages where articles exist. The crawler stores the html pages without any other element of the web page (images, css, javascript are omitted). By storing only the html page, the database is filled with pages that are ready for input to the 1st level of analysis, during which, our system isolates the ''useful text'' from the html page. The useful text can be defined as the title and the main body of the article. Information about this procedure can be found in [5]. The second analysis level receives as input XML files that include the title and body of articles. Its main scope is to apply pre-processing algorithms on this text and provide as output keywords, their location into the text and the frequency of their appearance in the text. These results are necessary in order to proceed to the third analysis level. Information about our pre-processing mechanism can be found in [4]. The core of our mechanism is located in the third analysis level, where the summarization and categorization subsystems are located. Their main scope is to characterize the article with a label (category) and produce a summary of it. All these results are then presented back to the end users of our personalized portal. The role of the portal is to feed each user only with articles that the user ''wants'' to face, according to his/her dynamically created profile.

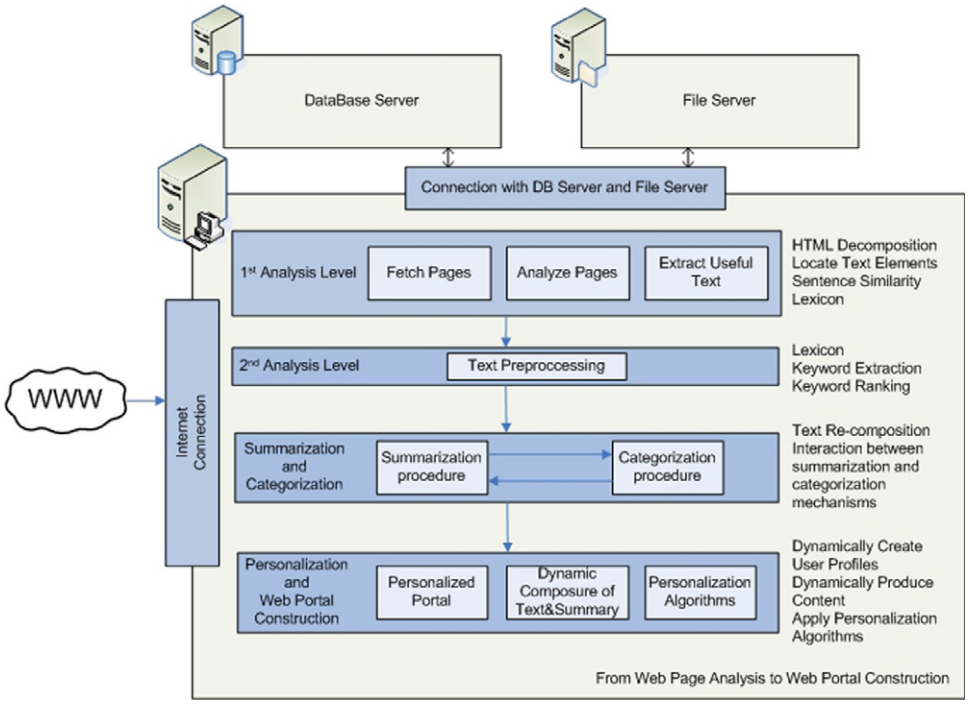


Fig. 1. System's architecture.

5. Algorithmic aspects

In order to analyze how each algorithm is applied on the texts, we will present an overview of the execution procedure. Algorithm 1 represents the execution of the procedures.

Algorithm 1 Label_article()

```
String Text = fetch_next_text();
List kwfr(text) = create_keyword_frequency_list(text);
List *kwfr_cat(category) = create_keyword_frequency_list(text,category);
Categorize (kwfr(text), *kwfr_cat(category));
if !Categorize then
    String Stext = Summarize(text,kwfr(text));
    List kwfr(Stext) = create_keyword_frequency_list(Stext);
    List *kwfr_cat(category) = create_keyword_frequency_list(Stext, category);
    Categorize (kwfr(Stext), *kwfr_cat(category));
    if !Categorize then
        Category = "generic";
    end if
else
    Summarize(Category,Personal_Data);
end if
```

Despite the fact that the procedure depicts only the labeling (categorization) of the articles, finally we achieve the three basic goals of the system: categorization, summarization, and interaction between the two afore-mentioned procedures. We start by trying to categorize the article and we create a list of the representative keywords (stemmed) of the text together with their frequency.

Next, we create identical lists for all the categories that reside in the database. These lists consist of the same keywords followed by their frequency in the category. We examine the cosine similarity of these lists in order to determine the category of the text (Table 1).

If the text cannot be labeled clearly, then we forward it to the summarization mechanism and check if the summarized text is able to be labeled. A text is supposed to be labeled whenever the cosine similarity is over a threshold and whenever the difference between the cosine similarity of the higher category and the others is more than a threshold. This will be explained thoroughly in the next chapter. Finally, if the cosine similarity between the text and the representative category is very high and the difference between the similarities of the other categories is enormous, then the text is added to the dynamically changing training set. The afore-mentioned procedure is also depicted as a block diagram in Fig. 2.

5.1. Summarization mechanism

5.1.1. Description

The summarization procedure is based on heuristic methods. This means that the summary is not constructed "from scratch", but it consists of the most representative sentences. This implies that every sentence should be given a score which leads to the construction of the summary. In the proposed mechanism, five distinct factors are used in order to create the summary and achieve the interaction with the categorization mechanism: (a) the keywords frequency (how many times a keyword appears in a sentence), (b) the keywords appearance in the title, (c) the percentage of keywords in a sentence, (d) the percentage of keywords in the text, (e) the keywords ability to represent a category, and finally (f) the keywords ability to represent the choices and needs of a unique user or a category of users with the same profile. According to the first two factors [(a) and (b)], we produce the first and basic equation to begin with a generic scoring of the sentences:

$$S_i = \sum w_{k,i}(k_1 + k_2) \tag{1}$$

Where $w_{k,i}$ is the frequency of the kth keyword of sentence i, k_1 is a constant that represents the impact of factor (a), and k_2 is a constant that represents the impact of factor (b) to the summarization procedure.

Table 1
Similarity between text and category

Keyword	Frequency
Business	0.742862
Entertainment	0.449297
Health	0.532352
Politics	0.418447
Integer	0.596509
Science	0.526925
Sports	0.642862

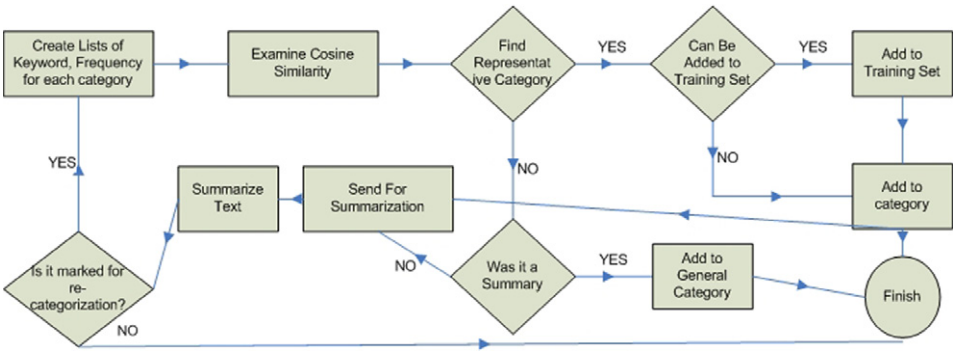


Fig. 2. The block diagram of the system's procedures.

5.1.2. Analysis

Through experimental procedure we have concluded to values for k_1 and k_2. k_1 derives from the following equation:

$$k_1 = 1 + 0.1x \tag{2}$$

where x is the times that the keyword is found in the title. Accordingly, k_2 derives from the following equation:

$$k_2 = 1 + 1.2y \tag{3}$$

where y is the possibility that the keyword is found n times in the sentence. Assuming a sentence with length m (m keywords) and a text with length t, factor y derives from the following equation:

$$y = \frac{n}{t}\frac{m}{t} = \frac{nm}{t^2} \tag{4}$$

In order to normalize the values that derive from Eq. (1), we propose the use of the factors (c) and (d). The normalization is needed as the big in length sentences tend to score higher than the small in length ones. The first represents the percentage of keywords in a sentence while the second represents the percentage of keywords in the text. More specifically, if three keywords are extracted from a sentence which consists of five keywords and the number of extracted keywords is 25, then factor (c) equals three of five (=3/5) and factor (d) equals three of 25 (=3/25).

The normalization we mentioned before is used in order to solve some problems that arise, like in the following example. Assume that a text has many small sentences and one which is very large. Additionally, the large sentence consists of 20 keywords and the extracted (useful) are five, while a small sentence that is very representative of the text consists of four keywords all of which are extracted as useful. The total number of useful keywords that are extracted is 30. The big sentence is more likely to score higher according to the aforementioned equation, as its length "helps" it to have more keywords. The two factors "normalize" this possible unfairness. The big sentence will have 5/20 and 5/30, respectively, while the second sentence will have 4/4 and 4/30 as (c) and (d) factors, respectively. In this way, the small in length sentence will be treated as more important than the big sentence. The normalization is applied directly to Eq. (1) and $S_i' = S_i N$, where N is the normalization factor and equals to the product of (c) and (d) factors.

The factors (e), keyword's ability to represent a category, and (f), keyword's ability to represent the choices of a unique user, are presented thoroughly in the following sections, as their influence to the procedure is important and promotes the summarization system into a fully personalized mechanism.

5.2. Categorization mechanism

5.2.1. Description

The categorization subsystem is based on the cosine similarity measure, dot products and term weighing calculations. More specifically, the system is initialized with a training set of articles collected from major news portals. The articles are pre-categorized – by humans – and are presented categorized into the news portals. Our training set consists of these pre-categorized articles. The categorization module receives as input the extract of the pre-processing mechanism. This is (a) an XML file containing stemmed keywords, their absolute frequency and their relative frequency in the article, and (b) the XML file containing the article (information about the article includes id, type, title, and body). After the initialization of the training set, the categorization module creates lists of keywords that are representative of a unique category, consisting of keywords with high frequency in a specific category and small or zero frequency for the other categories. The creation of the lists is helpful for categorizing newly arriving articles but we can prove that it can be helpful for summarization also.

5.2.2. Analysis

As the summarization procedure of our module is based on the selection of the most representative sentences which are selected by weighting them appropriately, the categorization outcomes can be helpful in adjusting more effectively the weighting of the sentences. Common sense implies that a keyword that has very high frequency for a specific category, should give more weight to the sentence in which it appears, while a keyword that has small or zero frequency for a category could add less to the weight of a sentence. Moreover,

a keyword that is included into the extracted keywords of an article that is representative of a category other than the one that the article is in, would give negative weight to the sentence. Eq. (5) is used for calculating the impact of the categorization into the summarization procedure

$$k_3 = \begin{cases} A \cdot cw_i & \text{where A} > 1 \text{ and } cw \text{ the positive category weight} \\ -A \cdot cw_i & \text{where A} > 1 \text{ and } cw \text{ the negative category weight} \\ 1 & \text{for neutral or not ranked by the system keywords or if } A = 0 \end{cases} \quad (5)$$

Parameter A must be greater than 1 and it is used in order to add a weight for the k_3 variable. If we want the summarization procedure to be based mainly on k_3, then weight values for A are used, but if the summarization should be equally based on all the "k" variables, then A should not be greater than the values that are assigned to k_1 and k_2. The parameter cw depicts the relative frequency of the keyword in the category. The relative frequency of a keyword in a category can provide us with evidence about how important the keyword for the category is.

With the use of Eq. (5), Eq. (1) is formed as shown below:

$$S_i = \sum w_{k,i}(k_1 + k_2)k_3 \quad (6)$$

5.3. User's role in summarization

5.3.1. Description

The personalization procedure of the portal, that is supported as a medium of communication between all the procedures and the users, can be used in order to personalize the summarization on each user. We believe that the user should be able to see a summarization of the articles that match his/her criteria and not a generic summarization that derives from a simple algorithmic procedure.

According to the algorithmic procedures of the personalized portal, the system creates lists of keywords for each user that represent his selection while browsing the news portal. More specifically, the keywords form two types of lists: a "positive" list with keywords that seem to suit the character of the user or a group of users, and a "negative" list with keywords that are out of interest for a user or a group of users. These lists derive from the selections of the user (which articles the user selected to read and which he/she did not, in which articles the user spends more time to read and in which he/she does not, etc.). Our intention is to rank higher the sentences which include "positive" keywords and to lessen the rank of sentences that include "negative" keywords for the user. In this scope we add another "k" variable, k_4, which will act as the personalization factor.

5.3.2. Analysis

The personalization variable is used like the variable that derives from categorization, and is given by the following equation:

$$k_4 = \begin{cases} B \cdot uw_i & \text{where } B > 1 \text{ and } uw \text{ the positive user's weight} \\ -B \cdot uw_i & \text{where } B > 1 \text{ and } uw \text{ the negative user's weight} \\ 1 & \text{for neutral or not ranked by the user keywords or if } B = 0 \end{cases} \quad (7)$$

The parameter uw depicts the relative frequency of the keyword for the user. The relative frequency of a keyword in a category can provide us with evidence about how important is the keyword for the user. This variable is added as a product to Eq. (6) which is formed as follows:

$$St_i = \sum w_{k,i}(k_1 + k_2)k_3k_4 \quad (8)$$

The variables A and B in Eqs. (6) and (7), respectively are used in combination to each other. If we do not intend to use one of the categorization or the personalization factors, then we may set the A or B variables to 0 for the omitting factor. If we want to focalize mainly on the personalization factor and less on the categorization, then we can set $B = 2$ and $A = 1$. This means that k_4 factor will have twice the impact of k_3. Table 2 shows the impact of (e) and (f) factors according to values of A and B.

Table 2
Impact of A and B to sentence weighting

A	B	Result
0	0	Personalization and categorization factors not computed to the result
0	1	Only personalization factor has impact to sentence weighting
1	0	Only categorization factor has impact to sentence weighting
1	2	Personalization factor has twice the impact of the categorization factor to the result
1	10	Personalization factor is so bigger than categorization factor that the impact of the categorization factor is almost not observed
1	1	The same impact for personalization and categorization factor
1.2	1.8	The values used for the evaluation of the mechanism

Table 3
Reaction of summarization algorithm to variables k_3 and k_4

Variable k_3	Variable k_4	Result
Positive	Positive	Positive
Positive	Negative	Negative
Negative	Positive	Positive (k_3 not computed to the result)
Negative	Negative	Negative

As observed from Eq. (8), some "special" occasions may occur from the negations that are introduced by the variables k_3 and k_4. Table 3 shows the reaction of the algorithm to the four different states.

One "special" occasion occurs when the categorization variable is negative and the personalization variable is positive. In this occasion we assume that the user, despite the fact that the keyword is not concerned as a representative of the category, has selected the specific keyword as a representative of his interests and thus the personalization variable overrides the categorization variable. Additionally, when both variables are negative the result remains negative, as the negations in our situation mean even lower score for the sentence.

6. Evaluation and experimental results

6.1. Evaluation

Each of the afore-mentioned Eqs. (1), (6), and (8) for sentence weighting was tested on some pre-summarized (by humans) texts. The results of our mechanism seem to be adequate compared to already existing mechanisms. Our main aim is to focalize on the personalized summary and thus the summaries that derive from Eqs. (1) and (6) may be less effective than the already existing algorithms. The personalization procedure into the summary cannot be evaluated by any prototype human created summary, because every human created summary implies the subjective human factor. The only evaluator of the system is the end user that receives the summaries. We tested our summarization algorithm compared to MEAD summarizer algorithm and the summarizer that is used by Microsoft Word. The personalization summaries are ranked by five test users who use the personalized portal.

6.1.1. Evaluating the automatic summarization mechanism

In order to ensure that the procedure before embedding the personalization factor produces adequate results for summaries, we evaluated our mechanism in comparison with results from Microsoft Word's summarizer. The results are compared to extracts from MEAD summarizer onto 30 articles from major USA and UK portals. The metrics that were used in order to calculate the results were precision and recall.

From the results derives that the summarization mechanism produces adequate results compared to tests that have been done with MEAD summarizer and obviously better results than the ones extracted by MS Word. By adding the categorization factor to the summarization mechanism, we manage to get slightly better results. We observe an overall increase of about 10% to the previous results concerning the metrics of precision and recall. The difference derives from the categorization procedure and, more specifically, from the addition

of k_3 factor to the summarization equation. This factor enables the higher ranking of sentences which include keywords representative of the category that the article belongs to. If an article does not include many keywords from the category to which it belongs, no changes occur. In this occasion, it is remarkable to note that after some time (in this time more keywords are inserted in the system) when someone tries to access the summary of the specific article it will be updated and the metrics of precision and recall will be measured higher than the first time of summarization. In the following table the metrics of precision and recall are presented for a specific article and how they change when new articles are categorized and more representative keywords for the category are inserted into the mechanism. The articles ''arrive'' in our system every 4–6 h as the major news portals update their data very often (see Tables 4 and 5).

From the previous statistics derives that the system is not static, but it is able to dynamically change and update the summaries that are extracted. Moreover, it is expected that after the publishing of an important news event, many articles on this issue will occur and will be published. This means that in the next 103 articles of the category that are captured by the mechanism within the next 78 h, at least one of them will be similar to the first article either as an update or as a complement. This derives also from the functionality of the modern news portals which include the ''related articles'' feature.

6.1.2. Evaluating the personalized summarization mechanism

The evaluation of a dynamically created personalized summary is not a procedure that can be completed comparatively. The measure that is used in order to evaluate the extracted personalized summaries is the relation between the summary and the article observed by the users of the mechanism. The procedure that was used in order to evaluate the results of the algorithmic procedure was: (a) provide the users with the full text of the article, (b) provide the users with both of the summaries created by using Eqs. (6) and (8), and (c) let them choose which summary they believe as more representative of what they read. The reverse procedure was also tested, which means first provide the users with both of the summaries, then the article and finally let them decide which summary they believe represents the most suitable for the full article they read. In both occasions the answers were the same.

Table 4
Comparison of summarization algorithm to MS word summarizer (results compared from outcomes of the MEAD summarizer)

	MS word		Proposed mechanism	
	Precision	Recall	Precision	Recall
Article 1	0.33	0.12	0.66	0.75
Article 2	0.12	0.25	0.75	0.66
Article 3	0.25	0.12	0.5	0.66
Article 4	0.25	0.12	0.75	0.5
Article 5	0.33	0.5	0.66	1
Article 6	0.33	0.25	0.66	0.75
Article 7	0.25	0.33	0.75	0.66

Table 5
Changes in precision and recall for the summary of article 1 after the addition of more representative keywords for the category that the article belongs

Time (after arrival)	Articles added to category (sum)	Proposed mechanism	
		Precision	Recall
10 min	0	0.5	0.66
8 h	8	0.5	0.66
24 h	31	0.66	0.5
36 h	43	0.66	0.66
48 h	59	0.66	0.66
62 h	88	0.75	0.75
78 h	103	0.75	0.8

The outcomes of the user's opinions can be separated into three groups: (a) new users of the system, (b) old users of the system but with little action (which means few data for personalization), and (c) advanced users of the system with high daily action (which means a lot of data for personalization). According to these categories, three different states were observed. The novice users noticed that the summaries were identical, which is a logical observation, as the system does not have enough information for the personalization procedure and thus, the sentence weighting for summarization is not affected by factor k_4 (used for personalizing the summary). The users of the second group selected in more than 80% of the cases the summary extracted from Eq. (6) (without the personalization factor). This was also expected as the dynamically created profile of such users (with low participation) was not complete and it included many keywords that were of low importance both for the article and its category. The most important results derive from the users of the third group. This group of users is considered to be advanced for the system with almost stable profiles after long time of system usage. The stability and completeness of the profile empowers the personalization procedure of the summaries. This group of users selected in more than 90% of the occasions the personalized summary as the most representative of the article according to their opinion and only 3% of the summaries were reported to be identical. It is important to note that most of the remaining 7% of the articles were reported to the categorization procedure of the mechanism as: ''belonging to a specific category but with weak connection''. This means that these were articles that added to the specific category with the ''note'' that the system had not managed to enclose them into a specific category but the category that they are inserted in, is the most likely to hold these articles.

6.2. Experimental results

Armed with our summarization and categorization mechanisms, we conducted experiments that would reveal the two-sided relationship between categorization and summarization. In order to have a working knowledge base (even a small one), we gathered news articles from some major news portals from the UK and the US. We defined six distinct news categories: business, entertainment, health, politics, science and sports and organized our captured texts (around 180 for each category) to them. Afterwards, using our categorization mechanism, we extracted 50% of the keywords of each text and associated each keyword with the text's category using the absolute frequency as a relativity measure. In particular, we carried out three types of experimental procedures. First of all, we needed to determine the text's keywords percentage we should keep, in order for our categorization module to be the most effective. Towards this direction, we modified the keeping percentage from 0.1 (i.e. 10% of the keywords) to 1 (i.e. all the keywords) with a step of 0.1, using a representative text for each of the afore-mentioned categories, and categorized it. The text that was entered to our categorization module had not been used for the construction of the knowledge base (i.e. it was not part of the training set). For each keyword percentage we measured the cosine similarity between the text and each category that resides in our knowledge database. We conducted the experiments using a minimum keyword size limit of five and six letters, both for the knowledge base and for the text that was about to be categorized. Following are some charts depicting the results.

From Fig. 3 (categorization procedure results), it is concluded that a percentage of 30% of the text's keywords should be kept for our categorization procedure to be optimal. Even though a lower percentage might be sufficient to decide on the text's category, we are keeping a percentage of 30% because, firstly it gives us almost always the right category decision and secondly, it provides us with a stronger distinction percentage between the correct category and all the others. In our opinion, this difference in similarity is the most important factor for a categorization mechanism, since it can provide us, even with expanding knowledge databases, with correct category answers. For example, it is possible when our database has many categories, some of which are similar to each other, the similarity of an input text to be relatively high to more than one category. In this case, the difference of similarity can be a better measure for categorizing, rather that an absolute similarity threshold.

It is clearly depicted in Fig. 4 that a text can achieve better scoring using a minimum keyword size limit of five letters and keeping 50% of the resulting keywords (from the training set). This way the knowledge base is more refined, while no category-important keywords are left out of the procedure.

In the next step of our experimentation, we wanted to examine the influence that the summarization procedure has on the categorization stage. In order to achieve this, firstly we summarized some humanly pre-cat-

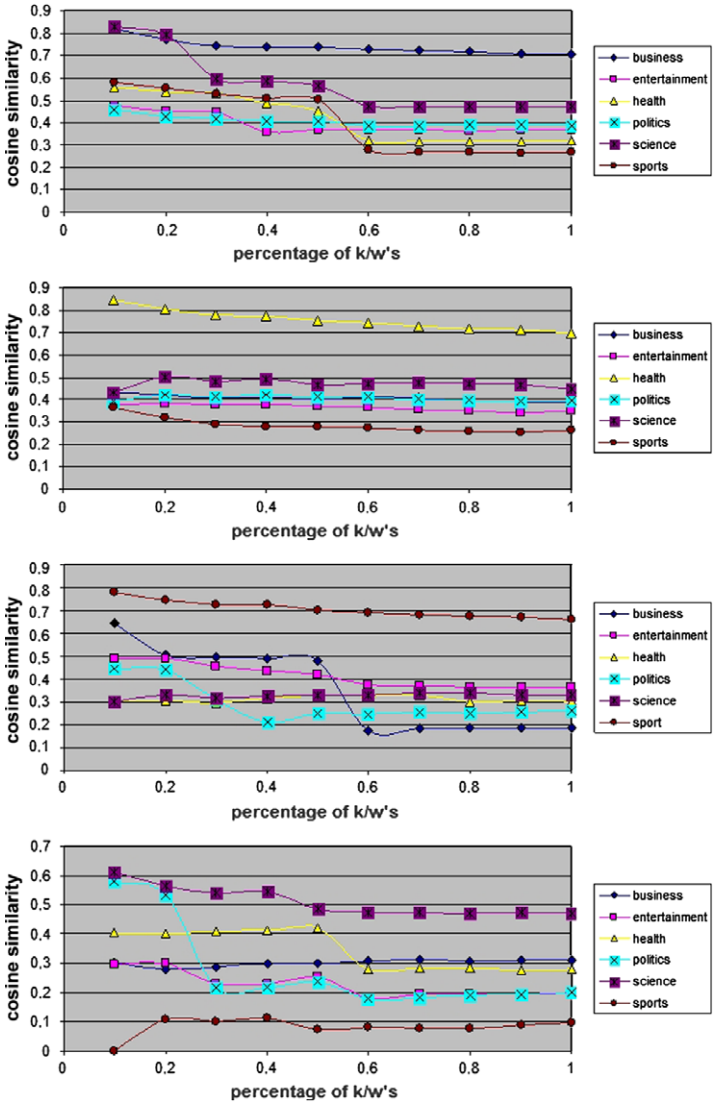


Fig. 3. Cosine similarity of texts compared to categories. Training set is constructed with 50% of the keywords kept (pre-processing procedure).

egorized texts and then inserted them into the categorization procedure. Finally, we compared the output of the categorization module (which in this way gives the summarized text's relativity with each registered category), with the pre-defined category of the text (see Figs. 5–7).

We used multiple summarization sizes in order to see the effect that they have on the categorization of the summary. Following are some sample charts of this experimentation that took place using texts belonging to different categories, which reveal the ideal percentage of sentences that could form a summary.

From this kind of experimentation we noticed that when keeping a plausible amount of the initial sentences, around 20%, for producing the text's summary, we could categorize the summary correctly to the text's category, thus saving a tremendous amount of work on the categorization side, since the summary is only a small portion of the text. This result is of huge importance for a fast responding, real time categorization system.

Another field that our experimentation investigated concerns the effect of the categorization to the summarization procedure. In order to discover the potential relationship, we constructed our summarization mech-

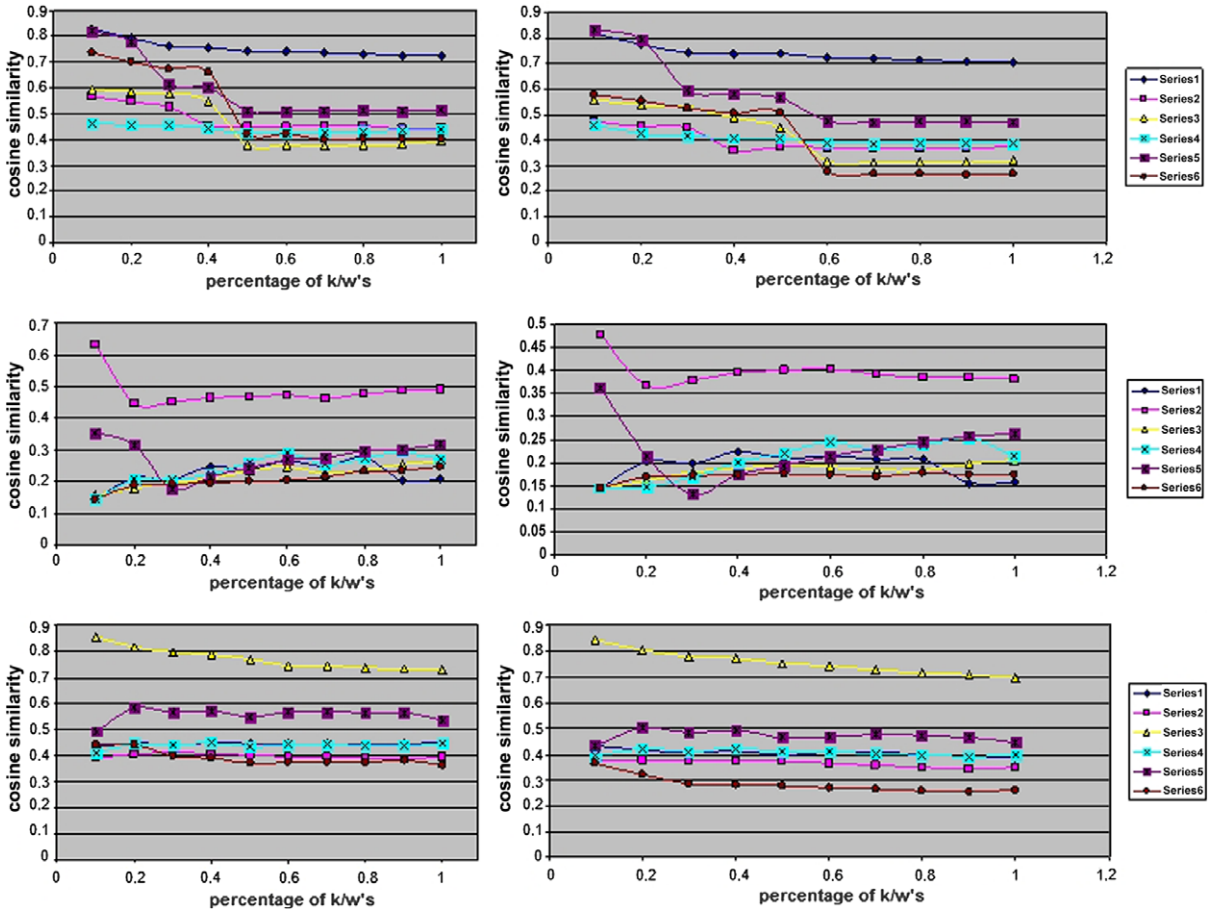


Fig. 4. The first column depicts the cosine similarity measured by utilizing the 50% of the keywords from the training set and the second column is the same cosine similarity measured by utilizing the 100% of the keywords from the training set.

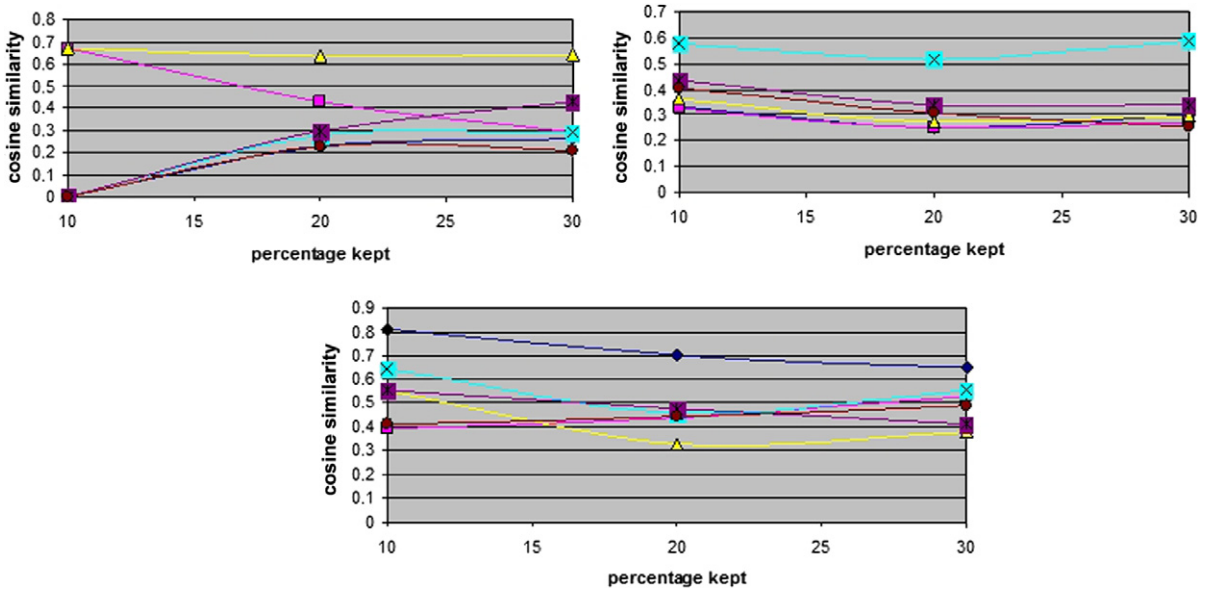


Fig. 5. Cosine similarity measured for categorizing summaries by keeping different percentages for creating the summaries.

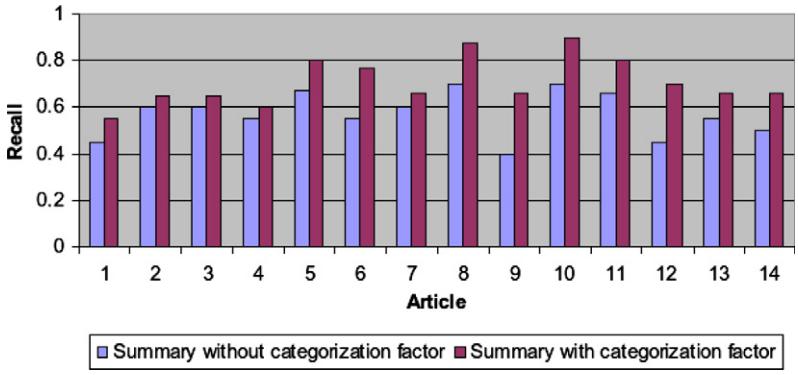


Fig. 6. Comparison of recall from summaries extracted with and without categorization factor.

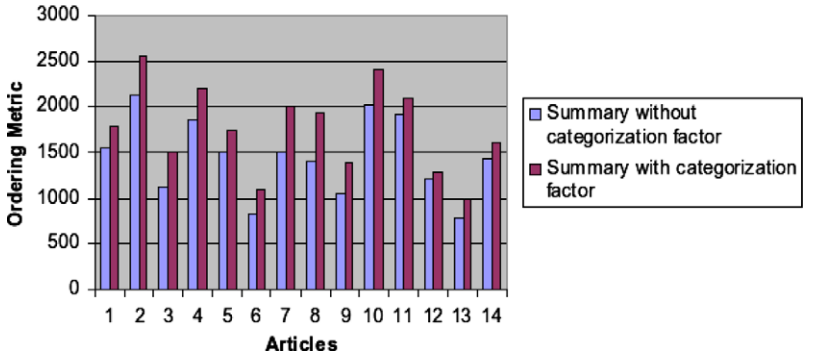


Fig. 7. Comparison of ordering metric from summaries extracted with and without categorization factor.

anism incorporating the categorization feature. For example, when we know a priori the text's category, this information is taken into consideration from the summarization module and each sentence rating is adjusted accordingly. For example, should a sentence contain many keywords irrelevant to the text's category (a priori knowledge), its rate will be much lower, or even negative, than when we don't know the text's category.

Using corpus texts, we first produced the text's summary without the use of the categorization factor (i.e. $k_3 = 1$) and afterwards we used this extra information to produce the summary and compared both of the results with the text's summary, which came with the corpus and was formatted by humans. The results are quite positive since we discovered that the categorization feature improved our summarization results by a factor of 10% or even more, meaning that the sentences which our summarization mechanism kept after the use of the categorization information are closer to the "optimal".

In order to compare the results from both cases (using the categorization information and not) we used the recall metric, i.e., how many of the sentences of the human-formed summarization where recovered by each procedure, and a sentence ordering metric. The latest was used to indicate the importance that the order of the sentences has in a summary. For example, it is possible that both of the summarization techniques achieve the same recall scoring but the ordering of the sentences is better in one of them. In fact, we observed that the summarization technique which utilizes the categorization information produces not only better recall scoring, but also higher sentence ordering score.

7. Conclusion and future work

We have presented a mechanism whose main aim is to combine summarization and categorization techniques in order to produce more efficient results for both the afore-mentioned mechanisms. The ultimate goal of the mechanism is to apply real time, efficient summarization and categorization which has proven to achieve well through the interaction of these subsystems. Since a major problem of today's Internet and,

more specifically, of today's news and articles streaming, is the burst mode that they are created in the Web, our intention is to collect as many of them for the users, refine them and present them back in a more humanistic manner. Our work focused on the core of the mechanism that we are creating, which is the categorization and the summarization subsystems.

We have proven that by using the outcomes of categorization we can achieve better results on summarization and vice versa. The algorithms used for the summarization procedure are based on heuristics, while the algorithm used for categorization is cosine similarity. The labeling of the articles achieves over 95 accuracy which is: achieving to categorize correctly almost all the articles into the prototype categories, while the results from the summarization mechanism are comparable to human created summaries. A major advantage of the system is that it manages to complete the whole procedure – from the fetching of the pages to the regeneration of the article to our portal – in less than 20 s per article. This means that the system is able to achieve real time regeneration of the articles.

We have also presented an algorithmic procedure that can be used in order to produce effectively personalized summaries. In an era of chaotic conditions in the web, personalization cannot be considered as a panacea but it can be very useful and helpful for advanced and novice users. In this scope we proposed a mechanism that is able to dynamically create summaries for texts or branches of text for the users to be able to view a summary that is fully personalized in their characteristic of browsing. This requires training for the mechanism which is based upon the selections and rejections of the users in the area of a web page and the time that he/she remained looking a specific web page.

The system that was described is generic and designed and constructed as a module. This implies that it can be embedded into software and mechanisms in order to extend them for supporting summarization procedures. Our main aim is to efficiently produce summaries for RSS readers and small screen devices. The last remark seems to be interesting and important as the usage of small screen devices for daily activities has reached a quite big number nowadays.

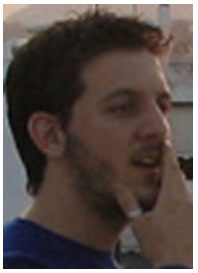
For the future versions of the core mechanism we will try to add a more complex algorithm for the creation of the summaries, though, since our scope is to create real time results we should be careful in order not to make a too complex system that requires long execution times. Additionally, what was observed was that, despite the fact that balancing factors were used, still, the greater in length sentences were gaining more weight than the shorter ones. Accordingly this implies that some short but inclusive sentence may be omitted. Furthermore, in order to globalize the system, some lexica should be included in order to make the pre-processing and summarization mechanism available for more languages than English. Finally, a crucial part of the mechanism is the implementation of the procedures for small screen devices. The ultimate goal is to use the mechanism in order to make PDAs, and generally small screen devices, more user friendly and available for daily tasks like reading mails, reading RSS feeds, and understanding the meaning of large amounts of text through a personalized summary. This mechanism could provide small branches of text to the users and let them choose easier which articles they are really interested in. Also, users could select the length of the summary they desire defining either a maximum of character length or a total amount of words.

References

[1] K. Ahmad, B. Vrusias, P.C.F. de Oliveira, Summary evaluation and text categorization, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003, pp. 443–444.
[2] R. Barzilay, M. Elhadad, Using lexical chains for text summarization, in: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, 1997, pp. 10–17.
[3] A.L. Berger, V.O. Mittal, OCELOT: a system for summarizing Web pages, in: Proceedings of the 23rd Annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, pp. 144–151.
[4] C. Bouras, C. Dimitriou, V. Poulopoulos, V. Tsogkas, The importance of the difference in text types to keyword extraction: Evaluating a mechanism, in: Hamid R. Arabnia (Ed.), International Conference on Internet Computing, CSREA Press, 2006, pp. 43–49.
[5] C. Bouras, G. Kounenis, I. Misedakis, V. Poulopoulos, A web clipping service information extraction mechanism, in: Third International Conference on Universal Access in Human–Computer Interaction, Springer, Las Vegas, Nevada, USA, 2005.
[6] H.P. Edmundson, New methods in automatic extracting, Journal of the ACM (JACM) 16 (2) (1969) 264–285.
[7] M. Fiszman, T.C. Rindflesch, H. Kilicoglu, Summarization of an online medical encyclopedia, MEDINFO (2004).
[8] U. Hahn, I. Mani, The challenges of automatic summarization, Computer 33 (11) (2000) 29–36.

[9] Eduard Hovy, Chin yew Lin, Automated Text Summarization in SUMMARIST, April 20 1997.
[10] W.L. Hsu, S.D. Lang, Classification algorithms for NETNEWS articles, in: Proceedings of the Eighth International Conference on Information and Knowledge Management, 1999, pp. 114–121.
[11] A. Jatowt, Web page summarization using dynamic content, in: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 2004, pp. 344–345.
[12] J. Kupiec, J. Pedersen, F. Chen, A trainable document summarizer, in: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, pp. 68–73.
[13] H.P. Luhn, The automatic creation of literature abstracts, IBM Journal of Research and Development 2 (2) (1958) 159–165.
[14] I. Mani, M.T. Maybury, Advances in Automatic Text Summarization, MIT Press, 1999.
[15] J.J. Pollock, A. Zamora, Automatic abstracting research at chemical abstracts service, Journal of Chemical Information and Computer Sciences 15 (4) (1975) 226–232.
[16] G. Salton, A. Singhal, M. Mitra, C. Buckley, Automatic text structuring and summarization, Information Processing and Management: An International Journal 33 (2) (1997) 193–207.
[17] M. Saravaman, P.C. Reghu Raj, S. Raman, Summarization and categorization of text data in high-level data cleaning for information retrieval, Applied Artificial Intelligence 17 (5) (2003) 461–474.
[18] D. Shen, Z. Chen, Q. Yang, H.J. Zeng, B. Zhang, Y. Lu, W.Y. Ma, Web-page classification through summarization, in: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval, 2004, pp. 242–249.
[19] J. Steinberger, K. Jezek, Using latent semantic analysis in text summarization and summary evaluation, Proceedings of ISIM04 (2004) 93–100.
[20] M.J. Witbrock and V.O. Mittal, Ultra-summarization (poster abstract): a statistical approach to generating highly condensed non-extractive summaries, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 315–316.

Christos Bouras obtained his Diploma and PhD from the Department Of Computer Engineering and Informatics of Patras University (Greece). He is currently an Associate Professor in the above department. Also he is a scientific advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Telematics and New Services, QoS and Pricing for Networks and Services, e-Learning Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published 250 papers in various well-known refereed conferences and journals. He is a co-author of eight books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. Also he is member of experts in the Greek Research and Technology Network (GRNET), Advisory Committee Member to the World Wide Web Consortium (W3C), IEEE – CS Technical Committee on Learning Technologies, IEEE ComSoc Radio Communications Committee, IASTED Technical Committee on Education WG6.4 Internet Applications Engineering of IFIP, ACM, IEEE, EDEN, AACE, New York Academy of Sciences and Technical Chamber of Greece.



Vassilis Poulopoulos was born in Kalamata, Greece in 1982. In 1998 he participated in the local contest of the Hellenic Mathematical society(Thales) and he achieved his competence in the National contest(Archimedes) of the society above. In 2000 he entered the Computer Science and Engineering Department of Patras University (Greece). He obtained his diploma on July 2005 and he is a member of Research Unit 6 since December 2001. He is responsible for the management of the web site of RU6. His basic fields of interest are: Databases, Web Technologies, Web Page Fetching and Analyzing, Automatic Text Summarization and Categorization, Web Page Personalization. He has experience in programming languages and more specifically ASP, PHP and JSP language programming, HTML programming, C, C++ and Java programming and SQL. He has participated in three projects (ASP-NG, Broadband Promotion in the region of Western Greece, Sig-Glue). He is currently participating in the Games At Large project running by Computer Technology Institute (CTI), Patras, Greece.

Tsogkas Vassilis is a senior undergraduate student at the Computer Engineering and Informatics Department of Patras University (Greece). His basic fields of interest are: PHP (mysql), HTML programming, C/C++ and Java programming. His research interests are: Web Technologies & Web-data Integrating, Dynamic Processing of Web Content, Information Extraction, Web Content Summarization and Categorization, Web Site Construction – Personalization. He is currently participating in the Games @ Large project running by Computer Technology Institute (CTI), Patras, Greece.