Creating Dynamic, Personalized RSS Summaries

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

[1]Research Academic Computer Technology Institute, N. Kazantzaki, Panepistimioupoli and Computer Engineering and Informatics Department, University of Patras, Greece
{bouras,poulop,tsogkas}@cti.gr

Abstract Automatically generated, human-quality text summarization systems are difficult both to develop and to evaluate, partly because articles differ along several dimensions: length, writing style and lexical usage. In this paper we propose a framework that, by utilizing RSS feeds, is able to personalize on the needs of the users and on the needs of their device, in order to present to the end-user only a fraction of the news articles covering just the useful information that derives from them. The created summaries utilize a weighted combination of statistical and linguistic features which leads to sentence scoring and selection. The procedure is assisted by categorization results as well as personalization algorithms that enhance the summarization module. The mechanism is evaluated using classic precision-recall metrics together with statistical results from real users. Within this framework we have created the PeRSSonal system that is able to create personalized, pre-categorized, dynamically generated RSS feeds focalized on the end user's small screen device.

Keywords: Text Preprocessing, Keyword Extraction, Personalized Summarization, Text Categorization

1 Introduction

During the last years, the technological advances of the World Wide Web have changed dramatically the ease of access to information. This change has also affected the manner and the frequency that news articles are created and published on the Internet. Every day, thousands of articles are created by the vast amount of news portals, major or minor, that exist in the WWW. This sense of freedom that the Internet inspirits is attracting more and more users, not just to read in a daily basis their "Internet newspaper", but also to create their own articles or their own sources of news articles. Besides, the latest "blogging" trend is not only targeted on publishing a personal dairy, but also acts as a medium of information exchange.

The aforementioned facts generate a number of repeated problems for the users of the internet who try to access information via their mobile phones, PDAs and generally small screen devices. These kinds of systems, that are becoming more and more common, already do have the power to run complex interactive applications [7]. However, their main problem lies on the useful space that their monitor has in order

to help users track and read articles. Despite the increasing resolution of PDA screens, limitations on the physical size of these will prevent the devices from ever reaching parity with the desktop [10]. The physical size of small screen devices, limits the maximum displayable content, which can be no larger than the dimensions of the machine in which it is embedded. On the other hand, the need for displayed text to be legible, defines another, more subtle boundary; if the size of text cannot be reduced below a threshold of legibility, then, as the screen shrinks in size, and less information may be shown on it, the user will be required to increase the level of interaction with the device in order to get to desired information. Our research work aims to deal with problems of this kind providing solutions that are device independent.

Conventional IR systems rank and present documents based on measuring relevance to the user's query. Unfortunately, most of the times, not all of the retrieved articles are of interest to the user. Summarizing texts that match the user's interests, can escort the user either in determining if the article is of interest, or understanding the text's overall meaning. The generated summaries can be a) generic, giving an overall sense of the article's content, or b) query-relevant, which presents the content that is most closely related to the initial search query. Personalized summaries fall within the latter.

In this paper we present a summarization procedure whose main scope is to support Internet users that are interested in reading, on a daily basis, specific news categories and we focalize mainly on users with small screen devices. The challenge is twofold: we are not only locating the news articles that the user is interested in reading, but also presenting them in such a way that the user will be able to read the most representative parts of them. Within these limitations, we present a mechanism based on personalized RSS feeds utilizing dynamic creation of summaries.

The well-known RSS protocol, which is based on the XML language and is part of the Web 2.0 framework, helps users confront consolidated information from websites and especially news portals. It is adopted by almost all the major and minor news portals and generally by websites whose content is updated often. Its goal is to provide the users with a title and a summary of an article, or with an important fraction of information that was published within a website, and let the user decide whether (s)he wants to view the complete article or not. Despite the fact that creating dynamic RSS content is not a difficult procedure, most news portals are mis-utilizing them.

Based on the fact that Internet users are becoming familiar with this protocol, we are developing a system that is exploiting RSS feeds in order to present filtered information to users in a more structured manner than the RSS feeds already provided by the major portals. More specifically, our system collects news articles from major and minor news portals; pre-processing techniques are applied to the collected articles and then categorization and summarization algorithms are used in order to refine them. Additionally, we empower the mechanism with a personalization factor in order to include the end-user to the whole procedure and thus enabling the system to produce isolated RSS feeds (title and summary for the latest articles), for each user, according to his/her personal device and preferences. Categorization and personalization algorithms are used as a means of enhancement to the summarization procedure.

In order to determine the effectiveness of the proposed mechanism, we evaluated the summarization techniques using precision-recall metrics. Even though it is a commonplace that there are no objective criteria for determining if the resulting summary is the best possible, precision-recall metrics can give us an estimation as to whether summaries are satisfactory. The resulting RSS summaries where evaluated from a different perspective also: the ability to present adequate information to the end users according to the device that the user is utilizing. We recorded the users' feedback of the mechanism concerning the coverage of their choices and needs of the system responses.

The rest of the paper is structured in the following manner: section 2 presents the related work in the field of summarization as well as the utilization of the RSS protocol in the Web 2.0 context. The flow of information within the system is presented in section 3, while the algorithmic aspects are covered in the next section. We present a thorough evaluation of the mechanism in section 5 depicting the results. In section 6 we express the conclusions of our research accompanied with some possible future work on the field.

2 Related Work

The goal of summarization, as described in [15], is the generation of a summary out of one or more, usually related to each other, articles and hence easing the user from the tedious task of reading large texts. A summary [17] usually helps readers identify interesting articles or even understand the overall story about an event. At most of the times, the summarization approaches are based upon a "sentence level" [8], where each sentence is rated according to some criteria (e.g. important keywords, lexical chains, etc.). Some techniques [6] try to find special words and phrases in the text, while Hayes, et al in [11] compares patterns of relationships between sentences. Taking into consideration the length of the sentences or the word case has also been tested [12].

While some summarization techniques try to extract the most important sentences, as far as a certain measure is concerned, others attempt to generate the summary using a knowledge-based representation of the content or a statistical model of the text [14]. Recently [2], 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 these terms are for the summary.

Despite the extensive work in the field of summarization, little effort has been made towards the direction of combining summarization techniques with the RSS news transmitting channels. Almost all news feeds provided by news portals (e.g. Google News [9]) consist of a title and a couple of the first sentences of the article (if not just the first words), while systems that have been proposed, like in [16], do not address the origin of the problem; the combination of dynamic summaries and RSS feeds.

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 on which newsgroup a news article belongs (e.g. Google News [9]). The purpose of text categorization as viewed by Hayes et al [11], 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 [13],[1]. New articles are categorized to the pre-defined categories using some criteria which vary from one 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, science, etc., or fine-grained where many categories, which are frequently overlapping with each other, are introduced. Linear Least Squares (LLSF) [18], a multivariate regression model that is automatically learned from a training set of documents and their categories gives good results. In this method, the training data are represented in the form of input/output vector pairs where the input vector is a document in the conventional vector space model (consisting of words with weights), and output vector consists of categories of the corresponding document. By solving a linear least-squares fit on the training pairs of vectors, one can obtain a matrix of word-category regression coefficients and by sorting these category weights, a ranked list of categories is obtained for the input document.

3 System architecture

Four major collaborating and autonomous subsystems constitute our mechanism: (a) crawler, (b) text preprocessor, (c) summarization and categorization subsystems, (d) personalized user response. The interconnection between the distributed subsystems is based on open standards for input and output in order to obtain a universal protocol for information exchange. Fig. 1 depicts the architecture of the complete mechanism.

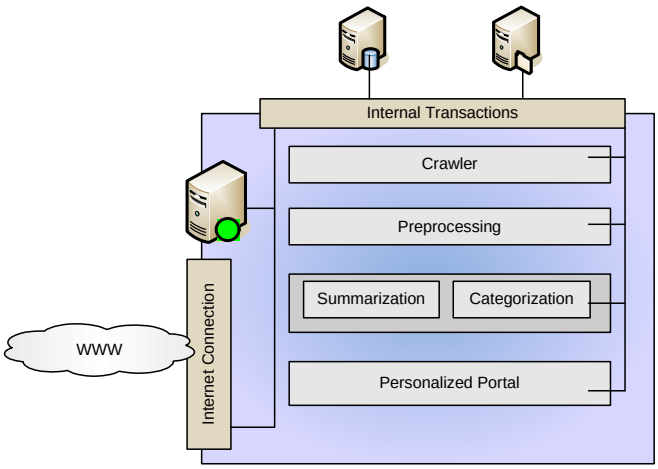

Fig. 1. General architecture of the mechanism

The procedure consists of the following steps: (a) capture pages from the internet and extract the useful text, (b) parse the extracted text and preprocess it, (c)

summarize and categorize the text and (d) personalize the results and present them to the end user using Web 2.0 protocols.

In order to capture the pages, a simple focused web crawler is used. The crawler receives as input the addresses that are extracted from existing RSS feeds, deriving from several major news portals. These RSS feeds point directly to pages where news articles exist. The crawling procedure is distributed across multiple systems which communicate with the centralized database. Crawled html pages are stored without any other element of the web page (images, css, javascript, etc. are omitted). During this analysis level, our system isolates the "useful text", which includes the title and the main body of the article, from the html page. More information about this procedure can be found in [3]. By storing only the useful text, as well as some other page meta-data, such as URL and insertion date, the database is populated with news articles that are ready for the text preprocessing step.

The second analysis level receives as input XML structured information, deriving either from the database or from raw XML files, which include the article's title and body. Its main scope is to apply text pre-processing algorithms on the article, resulting to output keywords, their location into the text and their frequency of appearance in it. These results are necessary in order to proceed to the third analysis level. Information about our preprocessing mechanism can be found in [4].

The core of our mechanism is located in the third analysis level, where the summarization and categorization sub-systems are located. The main scope of the categorization module is to assist the summarization procedure by pre-labeling the article with a category. This information is used internally by the summarization algorithms, as explained in the next section, providing better results. The outcome of the summarization procedure is further improved by the personalization module of our mechanism. Personalized summaries are finally presented back to the end users in the requested form (i.e. RSS feed). The role of the personalization layer is to feed each user only with summaries of articles that he/she "wants" to face according to his/her dynamically created profile, enhancing thus the summarization results.

4 Algorithm Analysis

In order to analyze how each algorithm is applied on the texts we will present the procedure that is followed in each step. The complete flow of information of our system is pictured in Fig.2

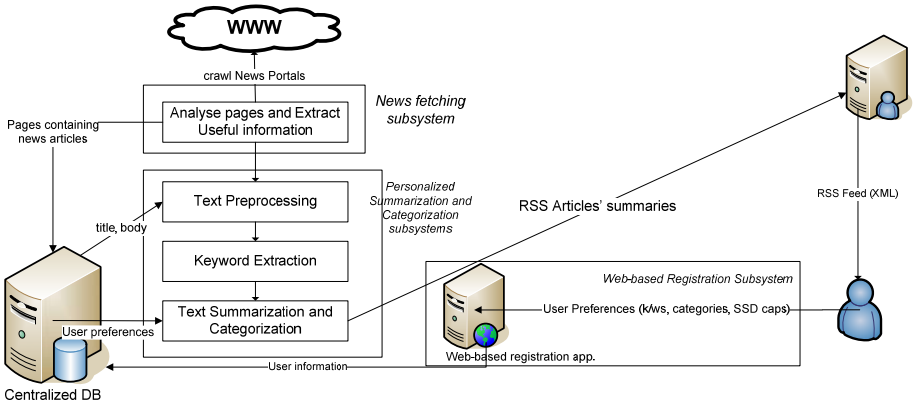

Fig. 2. Flow of information

3.1 Fetching articles and extracting useful information

The procedure of the News fetching subsystem, as depicted in Fig. 2, is: (a) capture pages from the WWW and analyze them extracting the useful text, (b) store the useful text in the centralized database. When a new article is fetched from the WWW, it is directed through our personalized text summarization and categorization subsystem, where its keywords are extracted and a categorization is attempted. At this stage, a default summary (non-personalized) is produced and stored in the DB, as well as all the aforementioned information, for future use.

Extracting the useful text from the HTML pages collected by the crawling mechanism is a procedure where the fetched web page is analyzed and the contiguous parts of it, which include a large amount of text, are considered to include useful information. Information about the "useful text extraction" mechanism can be found in [4].

The extracts of the useful text sub-system are mainly the body of the text and maybe a representative title. These parts are processed by the text pre-processing sub-system. This mechanism is assigned with the task of "cleaning" the text and extracting the keywords. The outcomes of the pre-processing mechanism are: stemmed keywords, their frequency in the text and their position in the text. A throughout description of this subsystem can be found in [4]. The information described before is enough for the following IR subsystems of our mechanism in order to apply categorization and summarization algorithms on the text.

3.2 Categorization procedure

Despite the fact that categorization is presented as a unique subsystem, it is closely related to the summarization one as its scope is to assist the latter. The categorization subsystem is based on the cosine similarity measure, dot products and term weighing calculations. The system is initialized with a training set of humanly pre-categorized articles, collected from major news portals. The categorization module receives as

input the extract of the pre-processing mechanism. This is (a) an XML structured source containing stemmed keywords, their absolute frequency and their relative frequency in the article and (b) the XML structure that contains the article's 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 at a specific category, and small or zero frequency for the others. The creation of the lists is helpful for labeling newly arriving articles, but has also proven in [5] to be helpful for the summarization procedure too.

The categorization attempt of a recently fetched article resembles the LLSF method and proceeds as follows; the labeling of the articles is done by using the list of the representative (stemmed) keywords of the text together with their frequency created by the pre-processing mechanism. Next, we produce identical lists for all the categories that we own. These lists consist of the same keywords followed by their frequency into the category. In order to determine the category of the text we examine the cosine similarity of the text and the categories based on the aforementioned lists.

From the outcomes, three different results are possible: (a) the text is very representative of a category and can be added to its dynamically changing training set of texts, (b) the text can be labeled (categorized) as it is very similar to one category compared to others and (c) the text cannot be labeled clearly. If the case is the later, the text is forwarded to the summarization mechanism, producing a generic/non-personalized summary and checking if the summarized text is able to be labeled.

A text is categorized whenever: (a) the cosine similarity between the text and the category is over a threshold, and additionally (b) the difference in the cosine similarity between the highest ranked category and the rest is more than a certain threshold. After experimental procedure, we decided that the best suited threshold for (a) should be no less that 0.50 (50% similarity), and the best-suited category (b) should have more than 11% difference in similarity with any other possible labeling.

Last but not least, when the cosine similarity between the text and the representative category is very high, and the difference in similarities, between this category and the rest, is also big, then the text is added to the dynamically changing training set. Experimentation gave us the best suited similarity thresholds for them: 65% and 20% respectively.

3.3 Summarization procedure

If a text is not categorized, then an attempt for a generic summarization is made. During it, we utilize two metrics: (a) the existence of a keyword in the title and (b) the frequency of a keyword. We call these factors k1 and k2 respectively. A keyword with very high frequency in the text is considered to be representative of it and thus, any sentence that includes it can be considered as text-representative. Additionally, any keyword of the text that also exists in the title is marked as an important one, so the sentences that include it are more representative. k1 derives from the following equation:

$$k_1 = 1 + 0{,}1x \tag{1}$$

where x is the times that the keyword appears in the title. k2 derives from the following equation:

$$k2 = 1 + 1{,}2y \tag{2}$$

where y is the possibility of a keyword appearing n times in a sentence. Assuming a sentence with length m of a text with length t, the possibility of a keyword to appear n times is:

$$y = p(n \mid m) \cdot p(m \mid t) \tag{3}$$

Based on these heuristics, we create a summary which consists of the most representative sentences of the text. In order to determine these, we deploy a score for each sentence according to the factors k1 and k2. Assuming that the text T has s sentences where i = [1..s] and f keywords where k = [1..f], each sentence is assigned a score according to the following equation:

$$W_i = \sum (1 + rel(\, fr(kw_{k,i}\,)))(k_1 + k_2) \tag{4}$$

where rel(fr($kw_{k,i}$)) is the relative frequency of the keyword k in sentence i.

After creating a generic summary, we retry to achieve a categorization, as the summarized text is more refined and consists only of important sentences and not of the whole text, which may include sentences with keywords that are distracting the categorization procedure.

The procedure that is followed in order to summarize a text after a successful categorization differs from the aforementioned steps due to the fact that another factor is included in the scoring. This factor is the keyword's ability to represent the category to which the document belongs. As long as the text is categorized, we can utilize this factor in order to create a more efficient summary. The theory that we are relying on is that, if the text is categorized, then some keywords in the text that are representative of the text's category should exist. This information can lead us to the use of another factor, k3 that covers the ability of the keyword to represent a category. Assuming that the relative frequency of a keyword within a category is cf, k3 can be evaluated as:

$$k_3 = A \cdot (1 + cf_i) \tag{5}$$

where A is the "special weight" of k3 and is added in order to represent how much the computation of the sentence weighting will be relied on factor k3. After experimental procedure, we concluded that a best fitted value for A is 1.2. However, it can be set to 1 if we do not want to rely on the k3 factor, or it can be increased to 1.8 in order to

rely mainly on the k3 factor and actually omit the k1 and k2 factors. Values less than 1 and more than 1.8 produce unexpected results as in the first occasion k3 leads to lessening of the sentence weight while in the second case the result does not rely at all on k1 and k2 (they are omitted).

If a text's keyword does not belong to the category of the text, then k3 is set to 1. A procedure that is experimented at the time being is allowing k3 to get negative values by examining whether the keyword text belongs to a category other than the one of the text. In this occasion, we assume that the keyword is representative of another category and not the text's category and hence, the overall weight of the keyword's sentence has to lessen. With the use of k3, the overall weighting equation is depicted below.

$$W_i = \sum(1 + rel(\, fr(kw_{k,i})))(k_1 + k_2)k_3 \tag{6}$$

3.4 Web Interface

The Web-based registration and user's interface subsystem represent the initial interface between the whole mechanism and the end user. A user registers in the system providing information about i) his small screen device (device capabilities) and ii) his keywords' or categories' preferences. This information is stored in the centralized database and is used later at the personalized summarization procedure.

While registering, each user is prompted with the categories that exist in the mechanism and is asked to assign a rate to each category according to his/her preference. . The score varies from -5 to 5, where "-5" means "I don't like it at all" and "5" means "I like very much". By selecting zero (0), the user indicates his neutral statement against the respective category.

Relying on these selections, we can create a simple user profile. At first, we create a list of the categories that the user likes and the ones that (s)he does not. This can help us with an initial "cleaning up" when selecting which news articles the user is interested in. The user is not just prompted to select the "likes" and "dislikes", but he selects a weight for each category. By utilizing these data, we are able to create a more detailed user profile which consists of a list of keywords that includes those that the user likes and the those that (s)he dislikes, followed by a relative frequency. The creation of the profile is constructed with the help of the following algorithm.

```
For each (selection s) {
If (s!=0) {
Keyword_name_usr = select 20*s keywords
                   from category keywords
// the keywords used for categorization, summarization etc
Keyword_weight_usr = select (2*s*relative frequency)
                   from category keywords
// the same list as above
}
else {
Keyword_name_usr = select 10 keywords
                   from category keywords
Keyword_weight_usr = select relative_frequency
                   from category. keywords
}
Insert into user profile keyword_name_usr,
                   keyword_weight_usr
If exists
Update user profile set keyword_weight += keyword_weight_usr
                   Where keyword_name = keyword_name_usr
}
```

From the user selections, we choose 20*s keywords, where s is the user's selection, (if user chooses 4 we select 80 keywords) from the training set's list, ordering the list by keyword's relative frequency in descending order. Additionally, we select the relative frequency of these keywords multiplied by 2s (if the user chooses -3 and the keyword has relative frequency equal to 0.02 then we extract -0.12). In this way, we end up selecting what is needed for the personalization procedure:

- Many keywords from the categories that the user has selected with high score (either positive or negative) and few keywords from the categories that the user has selected with low score.
- High positive value for the relative frequencies of the keywords belonging to categories that the user has selected with high preference, and low negative value for the frequencies of the keywords belonging to categories that the user has selected with negative preference.

These measures can help us refine the results presented to the user. By utilizing this information we can achieve the following:

- Select texts from the categories that the user likes and do not belong to a category that the user dislikes.
- Refine the outcomes of the summaries by adding the personalization factor.

The aforementioned procedure, gives us the ability to add another factor that is used for creating personalized summaries. The factor utilized is called k4 and can be used as a product to equation (4) or (6).

Assuming that for a user we have constructed a list of keywords followed by their relative frequency (preference of the user), k4 derives from the following equation:

$$k_4 = B \cdot (1 + uf_i) \tag{7}$$

where ufi is the user's preference for the keyword i and B is the "special weight" of k4 and defines how much will k4 affect the result of the sentence weighting. After experimental procedure we have concluded to the value 1.8 for B. When we have knowledge of an article's category, we apply the k4 factor on equation (6), while when we cannot categorize, we apply the factor on equation (4) as shown below:

$$W_i = \sum (1 + rel(\, fr(kw_{k,i}))) (k_1 + k_2) k_3 k_4 \tag{8}$$

$$W_i = \sum (1 + rel(\, fr(kw_{k,i}))) (k_1 + k_2) k_4 \tag{9}$$

The two-stepped refinement of the articles described earlier, is very helpful to decide, firstly on which articles to present to the end-user, and secondly, on the manner that the articles will be presented to the specific device of the user. This way the system has the ability both to select which articles to present to the user, relying on the his/her preferences, personalizing thus a dynamically created RSS feed, and to personalize the created RSS feed on the end-user's device, transferring only the amount of data that can be viewable within one or two pages of the specific small screen device.

5 System Evaluation

In order to evaluate the summarizer of the proposed mechanism, we followed an extensive experimentation and comparison of it with some well-known text summarization systems. In this scope, we created a user profile with high preference in the "business" category and low to the others. Next, we randomly collected 40 articles, which seemed to be relevant to the "business" category (as far as their in-portal categorization is concerned), from various news portals and categorized them. Afterwards, we examined the precision and recall outcomes when these articles are fed both to our system and to the MS Word and MEAD summarization mechanisms. The results are depicted in the following graphs (Fig.3,4).

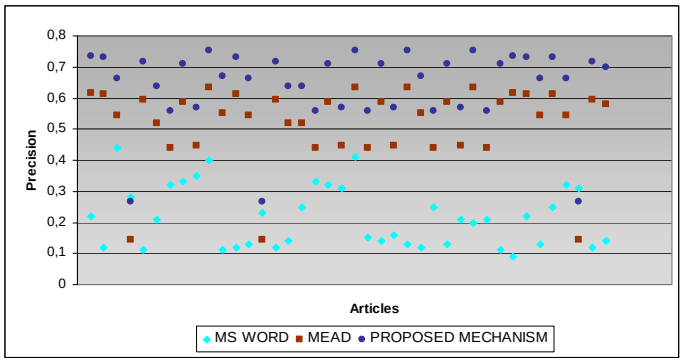

Fig. 3. Precision comparison between the proposed mechanism and the MEAD and MS WORD summarizers

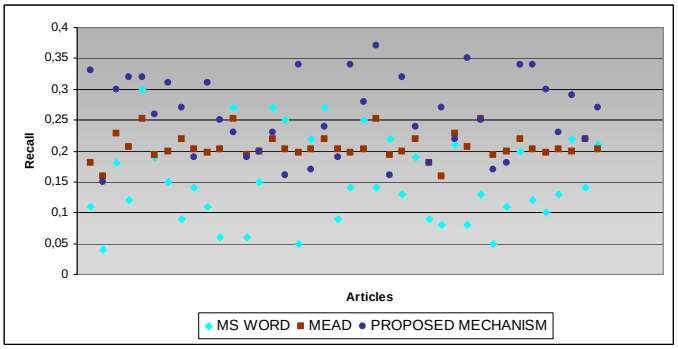

Fig. 4. Recall comparison between the proposed mechanism and the MEAD and MS WORD summarizers

From the previous graphs it is concluded that the proposed mechanism outperforms the MEAD summarizer by 23% and the MS Word summarizer by 200% as far as the precision metric is concerned. Recall results are quite similar: the proposed system recalls 24% better than the MEAD summarizer and 69% better than the MS Word summarization system. The above values are overall, meaning that there may still be some articles for which the proposed approach does not achieve better results. The explanation to this performance boost derives from two key facts; the summarization output produced by our mechanism takes advantage firstly of the pre-categorization done on the articles, and secondly of the personalization factor that is incorporated to the summarization procedure (summaries are personalized to the created user profile). In this way, the mechanism can achieve better scoring of keywords, and thus select more representative sentences, which in turn results in better summaries.

Following the evaluation of the core of the proposed mechanism, we tried to evaluate the system from a different perspective. When a new user arrives, (s)he provides his/her username and his/her screen capabilities. The later is auto-detected by the system (can also be user-modified) and is necessary in order to define i) the length of the news summaries sent back to the user and ii) the number of news articles

that are best suited for the device capabilities. A user also provides his category preferences, in the form of rating from -5 to +5, as well as any keywords that are of his/her special interest and should be highly rated through the article and sentence rating procedure.

When an unregistered user requests an RSS feed, a default RSS response, which contains the default summaries, is sent back. On the other hand, if the user is registered, he is fed with a personalized summary corresponding to his/her profile. The important factor to keep in mind is that different users receive different RSS responses, which vary in terms of news': length, ordering, amount, and categories. It is possible that two users receive the same articles but different summaries.

Apart from the obviously different responses of the mechanism under different circumstances, we needed to evaluate the positive effect that it had on the system's users. During this test phase, we created 10 virtual user profiles with specific preferences concerning the categories. We ensured that these people were receiving daily to their RSS reader the feeds from 10 portals and the feed of our portal (which collects articles from these 10 portals). We examined how many of these articles were of interest to the users, according to their profiles, in either of the cases.

From the following graphs (Fig. 5,6), it is clearly depicted that the mechanism presents an average of 85% less articles daily but the percentage of articles that the users seem to be interested in is more than 40% of the presented ones, while in the second occasion the users are interested in reading only the 7% of the articles presented. This means that the mechanism can achieve better clearing up of the feeds that the user is really interested in reading.

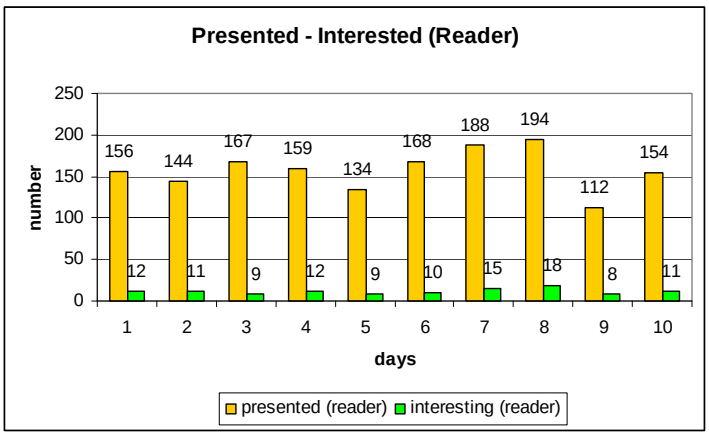

Fig. 5. Presented and Interesting articles directly from all news portals

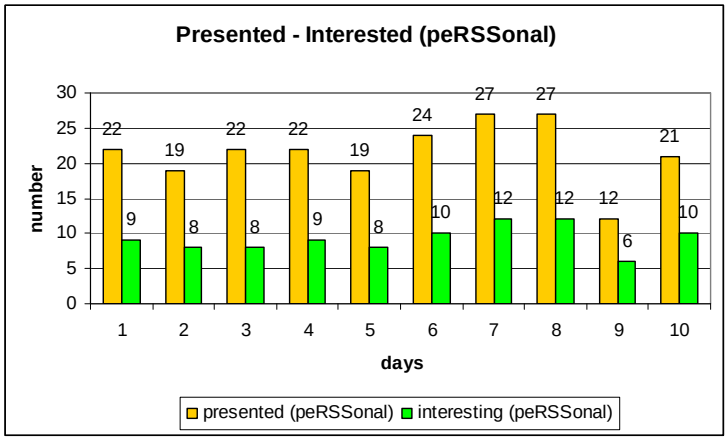

Fig. 6. Presented and Interesting articles from the proposed mechanism

6 Conclusion and Future Work

In this paper we presented a mechanism that is able to complete a procedure of collecting news from major news and present them personalized back to the end-users. This mechanism is extremely helpful for the internet users who are spending a lot of time trying to find news of their interest through major or minor news portals or even through RSS feeds and is already in evaluative use (PeRSSonal system). Despite the fact that the personalization micro-sites that exist even within some portals resolve part of the problem, still the refinement of the results and the personalization on the specific device of the user and the specific needs of the user is a huge problem.

The mechanism that we are proposing is able to collect the articles from news portals, categorize them, summarize them, and finally present them to the end-users according to their preferences and according to their device capabilities.

As future work for our mechanism we are considering a news tracker system which will be able to track the changes that are done on news articles. As more and more articles about a specific theme are published on several news portals or even on the same news portal, we should be able to collect all similar articles and present a summary of them back to the end user, providing also with the several links that the articles derive from and let the user make the best choice on which link to follow. Additionally, the automated procedure of collecting news articles must be empowered by a more effective focused crawler in order to avoid collecting unwanted data, putting the focus only on information that is needed as a feed for our mechanism. The keyword extraction mechanism could also be extended to include multilingual support with the use of lexica.

References

[1] Antonellis I, Bouras C. and Poulopoulos V. Personalized News Categorization Through Scalable Text Classification. In Proceedings of APWeb, (2006)

[2] Baron D. Persistent Media Bias. Stanford University, Graduate School of Business Research Papers: No. 1845, (2004)

[3] Bouras, C., Poulopoulos, V., Thanou A. Creating a Polite Adaptive and Selective Incremental Crawler. IADIS International Conference WWW/INTERNET, Lisbon, Portugal, Volume I, C., pp. 307 – 314 (2005)

[4] Bouras C., Poulopoulos V., Tsogkas V. The importance of the difference in text types to keyword extraction: Evaluating a mechanism. 7th International Conference on Internet Computing (ICOMP 2006), Las Vegas, Nevada. pp. 43-49 (2006)

[5] Bouras C., Poulopoulos V., Tsogkas V. Efficient Summarization Based On Categorized Keywords. The 2007 International Conference on Data Mining (DMIN07), Las Vegas, Nevada, USA, 25 - 28 June (2007)

[6] Ferragina P. and Gulli A. A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. In Proceedings of WWW Conference, (2005)

[7] Fitzmaurice, G., Zhai, S., Chignell, M., Virtual Reality for Palmtop Computers, ACM ToIS, 11,3, 197-218 (1993)

[8] Goldstein J., M Kantrowitz M., Mittal V., Carbonell J. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In Proceedings of ACM SIGIR Conference, (1999)

[9] Google News,. http://news.google.com

[10] Gutwin C. and Fedak C. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. Proceedings of the 2004 conference on Graphics interface, London, Ontario, Canada, 145 : 152 (2004)

[11] Hayes P.J., Knecht L.E. and Cellio M.J. A News Story Categorization System. In Proceedings of the second Conference on Applied Natural Language Processing (1988)

[12] Herman E. S. The Propaganda Model: A Retrospective. Against All Reason, 1: 1-14, (2003)

[13] Hsu W.L., Lang S. D. Classification Algorithms for NETNEWS Articles. In Proceedings of CIKM, (1999)

[14] Kummamuru K., Lotlikar R., Roy S., Singal K. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In Proceedings of WWW Conference, p.p. 658 – 665 (2004)

[15] Radev D., Otterbacher J., Winkel A. and Blair S. NewsInEssence: Summarizing Online News Topics. Communications of the ACM Vol. 48, No. 10, (2005)

[16] Wang, T., Yu, N., Li, Z. and Li, M. nReader: reading news quickly, deeply and vividly. Conference on Human Factors in Computing Systems 1385-1390, ACM Press New York, NY, USA, (2006)

[17] Wasson M. Using Leading Text for News Summaries: Evaluation Results and Implications for Commercial Summarization Applications. In Proceedings of ICCL, (1998)

[18] Yang Y. and Chute C.G. An example-based mapping method for text categorization and retrieval. ACM Transaction on Information Systems (TOIS), 12(3):252-277, (1994)