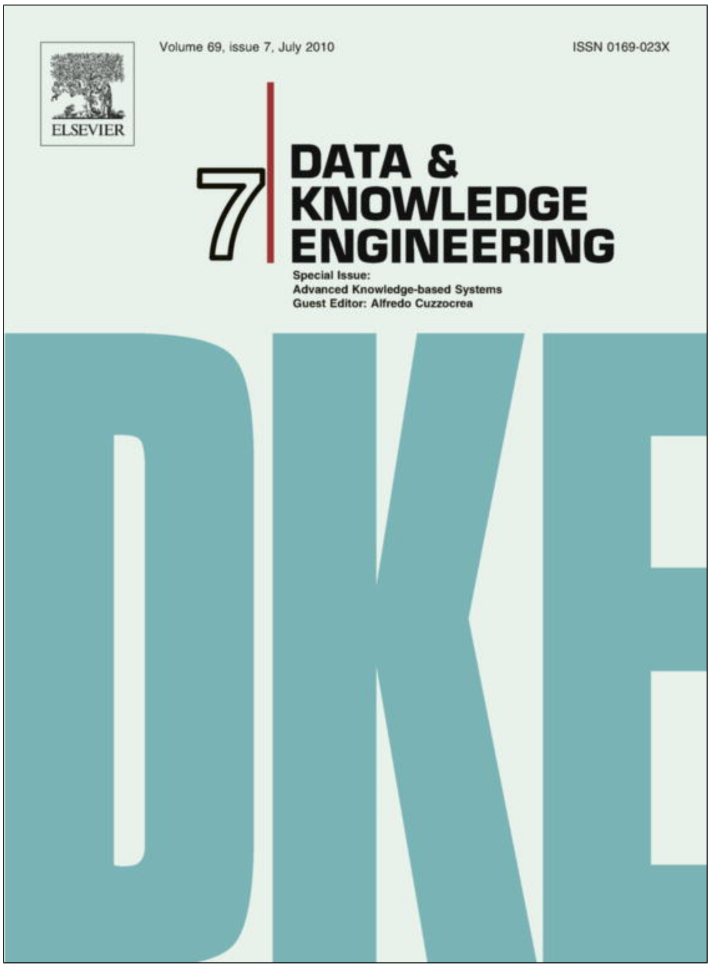

Volume 69, issue 7, July 2010
ISSN 0169-023X
ELSEVIER
7
DATA &
KNOWLEDGE
ENGINEERING
Special Issue:
Advanced Knowledge-based Systems
Guest Editor: Alfredo Cuzzocrea
DKE

Data & Knowledge Engineering 69 (2010) 664–677

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

# Noun retrieval effect on text summarization and delivery of personalized news articles to the user's desktop

Christos Bouras *, Vassilis Tsogkas [1]

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

ARTICLE INFO

*Article history:*
Available online 20 February 2010

*Keywords:*
Knowledge discovery
Web-based Information
Noun retrieval
Text summarization
News personalization

ABSTRACT

Text summarization and categorization, as well as personalization of the results, have always been some of the most demanding information retrieval tasks. Deploying a generalized, multi-functional mechanism that produces good results for the aforementioned tasks seems to be a panacea for most of the text-based, information retrieval needs. In this article, we present the keyword extraction techniques, exploring the effects that part of speech tagging has on the summarization procedure of an existing system. Moreover, we describe the profiling features that are used as an extension to an already constructed news indexing system, PeRSSonal. We are thus enhancing the personalization algorithm that the system utilizes with various features derived from the user's profile, such as the list of viewed articles and the time spent on them. In addition, we analyze the system's interconnection channels that are used with the client-side desktop application that was developed and we evaluate the approaches that we propose.

## 1. Introduction

The Web information age has brought a dramatic increase in the sheer amount of information, the access to this information, as well as the intricate complexities governing the relationships within this information. Nowadays, users tend to prefer personalized information that is easily delivered, without much hassle, to them. The aforementioned facts, probe for new interconnection architectures and transferring of data. Following this path, several interconnection protocols have risen in the last few years: XML/XSL [6], SAX [18] and DOM [5].

Based on the fact that many information retrieval (IR) tasks, such as classification, summarization or clustering rely heavily on keyword – oriented information originating from the source texts, it is easy to conclude that keyword extraction is a key step for them. In essence, keyword extraction, aims to select the appropriate keywords out of a text, accompanying them with a suitable score that depicts their importance. By appropriate, we mean the most representative words, as far as the text's overall meaning is concerned. Following the keyword extraction procedure, text summarization and categorization techniques come. Both of them operate on a keyword basis, working in a bipartite interactive way [3]. Moreover, user profiling serves another significant role in our system allowing a continuous feedback from the users, enhancing in a secondary stage the resulting information. In our research, a unified, yet autonomous system is developed, PeRSSonal [16], in which

* Corresponding author. Tel.: +30 2610 960375; fax: +30 2610 969016.
*E-mail addresses:* bouras@cti.gr (C. Bouras), tsogkas@ceid.upatras.gr (V. Tsogkas).
*URL:* http://ru6.cti.gr/bouras (C. Bouras).

[1] Tel.: +30 2610 996954.

0169-023X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2010.02.005

keyword extraction, summarization categorization and personalization are the core procedures. The system is stable enough for everyday use through its portal website [16] .

We are focusing on news articles served by the numerous portals from around the internet. This overwhelming amount of data deprives the users from easy access to unbiased, high quality information and this is the terrain of the proposed mechanism. We are thus researching the usage of multiple data mining and retrieval techniques which are incorporated to the proposed system. This is exactly the key feature that distinguishes the to-be presented approach from similar ones like [19,8,14]. The contribution of the current article is twofold: the evaluation of the effect that part of speech (POS) tagging has on text summarization, and the presentation of the applied personalization algorithm with regard to the content delivery to the user's desktop.

The rest of the article is structured as follows: in Section 2 we are highlighting some related work in the fields of interest. In Section 3 we give a brief description of the developed system. In Section 4 the architecture of the complete system is presented. Section 5 outlines the algorithm procedures of noun retrieval and personalization that are followed by the system. In Section 6 we present the evaluation results of the developed mechanism. Section 6 states the overall conclusions of this article, while Section 7 presents some thoughts for future additions to the system.

## 2. Related work

Automatic part of speech tagging is a well-known problem that has been addressed by several researchers during the last 20 years. It is a firm belief that when it comes to keyword extraction, the nouns of the text carry most of the sentence meaning. In a sense, extracted nouns should lead to better semantic representation of the text, and hence, improved IR results. Noun extraction, a subtask of Part of Speech (POS) tagging, is the process of identifying every noun (either proper or common) in an article or a document. In many languages, nouns are used as the most important terms (features) that express a document's meaning applied by natural language processing (NLP) techniques used widely in information retrieval, document categorization, text summarization, information extraction, etc. Various methodologies have been proposed making use of linguistic [12], statistical [4], symbolic learning knowledge [17] or support vector machines [9] and can be categorized to: morphological analysis, or POS tagging based. The former methods try to generate all possible interpretations of a given phrase by implementing a morphological analyzer or a simpler method using lexical dictionaries. It may over-generate or extract inaccurate nouns due to lexical ambiguity and shows a low precision rate. On the other hand, the POS tagging based methods choose the most probable analysis among the results produced by the morphological analyzer. Due to the resolution of the ambiguities, they can obtain relatively accurate results. However, they also suffer from errors not only produced by the POS tagger, but also triggered by the preceding morphological analyzer.

Personalization aims to customize the results on a user's explicit or implicit interests and desires. As explained in [15], the move to personalization is no longer an option, but a necessity. Several challenges however come into place for the personalization procedure to be successful; scalability, accuracy, evolving user interests, data collection and preprocessing, intergrading multiple sources of data, as well as privacy issues are only some of the aspects that our system faces up. From an architectural and algorithmic point of view, personalization systems fall into three basic categories: rule-based, content-filtering, and collaborative filtering systems.

Personalizing news feeds is an interesting subtask of news filtering and personalization that has emerged in the few last years. Its target is to effectively separate interesting news articles for a user from a large amount of documents. For example in [1], the authors make use of a machine learning classification framework to filter news following the user's choices. Another technique for adaptive news, which is based on user modeling, is presented in [19], where the system maintains separate user models for each topic of news. However these systems lack serving of news summaries and use fairly trivial keyword extraction techniques. Also, they expect from the users to explicitly modify their profiles which is often a tedious task. Another interesting approach is presented in [8] where the notion of information novelty is utilized. Despite the fact that the implemented algorithms perform well, they also lack automatic recording and prediction of user preferences which change frequently. An alternative approach is researched in [14] where the system produces multi-document summaries and has the ability to adjust to the various input texts. However this system lacks categorization and personalization features that could be combined with the resulting summaries, as in our work.

Presenting to the user summaries matching their needs is a very crucial procedure that can assist information filtering. Even though automatic text summarization dates back to Luhn's work in the 1950s [13], several researchers continued investigating various approaches to the summarization problem up to nowadays. A summary [20] usually helps readers identify interesting articles or even understand the overall story about an event. Most of the time, the summarization approaches are based on a "text-span level" [10], with sentences being the most common type of text-span having each of them rated according to some criteria (e.g. important keywords, lexical chains, etc.). These techniques transform the original problem to a simpler one: ranking sentences according to their salience or likelihood of being part of a summary, concatenating them at a second stage. Some techniques [7] try to identify special words and phrases in the text, while in [11] the authors compare patterns of relationships between the sentences.

Several text classification (categorization) approaches have been researched over the years: Naive Bayesian (NB), K-Nearest Neighbor (KNN), and Centroid-based (CB) techniques are some examples. Linear Least Squares (LLSF) [21], a multivariate regression model that is automatically learned from a training set of documents and their categories, gives good results and is utilized in our work.

In this article we present the incorporation of noun retrieval techniques in PeRSSonal, using the support vector machine (SVM) method for POS tagging, as part of its keyword extraction algorithms, and we explore, through experimentation, the possible improvements this change has on the mechanism's IR procedures: summarization and categorization. Furthermore, we are dealing with the effective and adequate presentation of personalized news summaries from articles that derive from the WWW to the user's desktop. We present the personalization algorithm that is used for presenting the pre-categorized and summarized articles to the user's desktop application which is capable of exchanging information with our already mature categorization and summarization system. Our personalization approach is mainly content-based with some collaborative filtering features by enhancing the algorithm with the ability to automatically adopt over time to the continuously changing user choices. Furthermore, we base our summarization procedure (enhanced by the personalization module) on the TF–IDF term-weighting model.

## 3. System description: PeRSSonal

PeRSSonal [16], the automatic summarization, text categorization, personalized syndication system, applies several data mining techniques through a layered infrastructure that achieves filtering of information and adaptability to the user. We focus on news articles that are gathered from numerous news portals from around the internet and we are targeting to the alleviation of the end-user from the cumbersome task of searching through this overwhelming plethora of information. PeRSSonal initiates a new sense of news exchange over the WWW, "meta-portals", which aims to consolidate and index news content from a multitude of sources. The proposed architecture however does not merely index and serve static content, but it incorporates surplus value by categorizing, summarizing and personalizing its content. Moreover with the use of the client-side module, the user is not anymore overrun by unnecessary content, keeping thus the network traffic to a minimum.

The user participates in the procedure by explicitly defining his/her preferences or by allowing the system to automatically adapt to his/her dynamically changing profile. This generates the notion of the "easily modified" user profile which is used extensively by the proposed approach.

## 4. Architecture

PeRSSonal follows a classic *n*-tier architectural approach. The system consists of four layers which work autonomously and collaborate through a centralized database. The procedure that is followed, as depicted in Fig. 1, starts with the interconnection between the mechanism and the web sources and it is where the primary tasks take place. These include: the content fetching procedure, the analysis of the downloaded content and finally the extraction of the useful information from the web content. In order to capture web pages, a simple focused web crawler is used. The crawler takes 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 and they also provide us with useful information per article (e.g. article's title) or per RSS (e.g. pre-categorized feed). The crawling procedure is distributed across multiple systems which synchronize thought the centralized database.

The various layers of the PeRSSonal server-side mechanism work autonomously and collaborate through the centralized database. Crawled HTML pages are analyzed and are stored without any other unnecessary page element (images, css, javascript, etc). During this analysis level, our system isolates the "useful text", meaning the main body of the article. 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 layer of the system, which is the focus of this article, operates on the article's title and body applying several preprocessing techniques. Text preprocessing is probably the most important preceding task of any text-based IR technique and hence our approach pays much attention to the various algorithms that are used. In particular, after the retrieval of the stored article that resides in the database, a series of inner procedures take place at this layer. Firstly, the article's language is recognized either directly through language identifying procedures, or indirectly using the predetermined language of the origin-feed. Following is a sentence separation and punctuation removal step. Afterwards, the noun identification step takes place which, by utilizing the POS SVM-based tagger [9], is able to determine with high precision the article's nouns. Some common text extraction techniques follow: stopwords removal and stemming. Noun extraction should precede these procedures if it is to succeed with high probability. It is important to note that the noun identification, stopwords removal and stemming procedures are language dependant, meaning that specific language rules, stopword lists and stemming rules respectively, have to be applied for different languages. The above set the foundations for multi-language support by our mechanism, even though only the English language has been incorporated so far. The results of the procedures described in this layer are stemmed keywords either marked as nouns or not, their location in the text and their frequency of appearance in it. These are represented through term frequency – inverse document frequency (TF–IDF) vector statistics that are stored in the database and are utilized by the procedures of

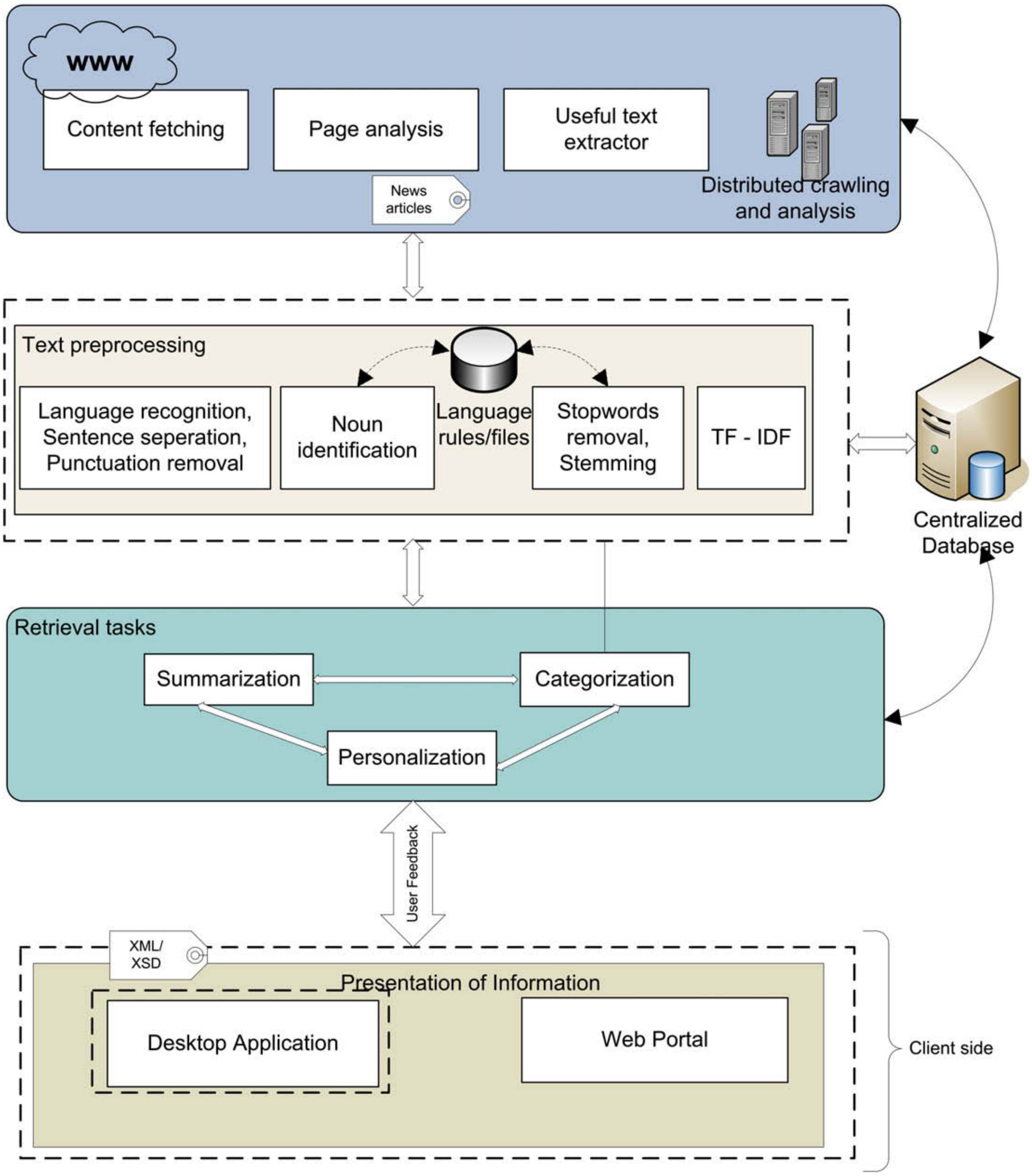


**Fig. 1.** The System's architecture.

the third analysis level. The aforementioned keyword extraction subtasks are to their majority, language dependant, meaning for example that different stopwords lists, stemming rules and noun retrieval algorithms are used for every different language.

The information retrieval tasks of our mechanism are located in the third analysis level, where the summarization and categorization algorithms are applied. The main scope of the categorization module is to assist the summarization procedure by pre-labeling the article with a category and has proven in [3] to be providing better results, as far as summarization is concerned. Following the IR tasks of the system, personalization algorithms take place. The personalization module that is also described in this article is easily adaptable to the user meaning that, small changes to the user's preferences as expressed by his/her browsing behavior are detected, adjusting thus the profile. Our personalization algorithm uses a variety of user-related information in order to filter the results presented to the user.

Finally, the content is delivered to the user using XML formatting for data content and XSD schemas for the transmitted data. The personalization procedure on the server-side, assisted by the feedback information originating from the user's behavior, as well as the delivery of the summarized content to the user's desktop through the client-side application are extensively covered in the current work.

## 5. Algorithmic aspects

Our analysis consists of the following different algorithmic steps: extraction of keywords and identification of nouns, categorization and personalized summarization procedures and finally, delivery and presentation of the information to the user's application.

### 5.1. Keyword extraction and noun identification procedure

The input to the keyword extraction module is plain text that defines the article's body and title as well as its language. Apart from the previous, some parameters have to be tuned in order for the mechanism to be the most efficient: (a) minimum word length (all words with length smaller than the minimum are removed) and (b) the language dependant stopword list that will be used. Our experimentation for news articles in [2] revealed that a limit of five letters is best suited as far as articles written in English are concerned.

Noun identification involves an off-line learning step for the POS tagger using language specific rules. Previous to the tagging, SVM models (weight vectors and biases) are learned from a training corpus using the learning component. A modified version of the SVMTool [9] is used so that tagging takes place only for nouns, saving system's processing time. Once the training is complete, the article's body is forwarded to the tagger and the text's nouns are marked. Stopwords removal takes place and stemming rules are applied, resulting in the TF–IDF vector for all the texts and their terms.

### 5.2. Categorization procedure

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 1500 pre-categorized articles, belonging to seven different basic categories. Those categories are the basic ones that are served by most of the news portals online. The amount of articles is enough to give an adequate knowledge base on which text classification can proceed. The categorization module receives as input the extract of the preprocessing mechanism, which is: (a) stemmed keywords, (b) noun-related information, (c) absolute and relative frequency of the keywords appearance in the article and (d) the article's title and body. After the initialization of the training set, the categorization module creates lists of keywords-nouns that are representative of a unique category, consisting of nouns with high frequency at a specific category, and small or zero frequency for the others.

The categorization attempt of a recently fetched article resembles the LLSF method and proceeds as follows: firstly we generate the list of the representative (stemmed) keywords of the text together with the frequencies evaluated by the preprocessing mechanism as depicted in Table 1. Following this we produce identical keyword-frequency lists for all the categories that reside in the database and which consist of the same keywords followed by their frequency into the category (Table 2).

In order to determine the text's category, we examine the cosine similarity of the text and the categories based on the aforementioned lists.

An article is most of the time related with a similarity measure to more than one category. However, for a categorization result to be accepted we define two thresholds: (a) the cosine similarity between the text and the category should be over $T_{\mathrm{hr1}}$, and (b) the difference of the cosine similarity between the highest ranked category and the rest of the categories should exceed $T_{\mathrm{hr2}}$. Experimentation, gave us the best suited thresholds for $T_{\mathrm{hr1}}$ and $T_{\mathrm{hr2}}$ as 0.50 (50% similarity), and 11% respectively. If $T_{\mathrm{hr1}}$ or $T_{\mathrm{hr2}}$ is not met, the article is forwarded to the summarization module and the resulting generic summary is used as input to a second categorization attempt for the article. Should (with the second categorization attempt) the above thresholds now be met, the labeling of the summary is kept, while at a different case, when neither the initial nor the secondary attempt satisfy the thresholds, the initial labeling of the article is kept.

### 5.3. Summarization procedure

During the summarization procedure, we utilize three factors: (a) the existence of a keyword in the title (b) the frequency of a keyword and (c) the noun tagging information of a keyword. We call these factors $k_1$, $k_2$ and $N$ respectively. A keyword

**Table 1**
Article's categorization vector.

| Stemmed k/w | Frequency |
|---|---|
| minist | 7 |
| Obama | 3 |

**Table 2**
Politics category vector.

| Stemmed k/w | Frequency |
|---|---|
| minist | 90 |
| Obama | 200 |

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. Moreover, when a keyword is tagged as a noun, we consider it significant thus boosting it with some extra weight. Parameters $k_1$ and $k_2$ are thoroughly explained in [3], where through experimental procedure we have concluded values for $k_1$ and $k_2$ based on statistical results. $N$ derives from the following equation:

$$N = L * z \tag{1}$$

where $z = 0$ if the keyword is not a noun and $z = 1$ if it is. $L$ conveys the desired extra weight that a noun existing in a sentence should have. Experimentation with various $L$ values revealed that $L$ should be no more than 1.5 or else sentences with few keywords-nouns receive low scores, compared to sentences with many nouns, and are substantially excluded from the summary. Typical values for $L$ range from 0 to 1 with the former depicting that the summarization algorithm is not taking into consideration the noun relevant information.

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 $k_1$, $k_2$ and $N$. Assuming that the text T has $s$ sentences where $i = [1..s]$ and $f$ keywords where $k = [1..f]$, each sentence $i$ is assigned a score according to the following equation:

$$W_i = \sum (1 + rel(fr(kw_{k,i})))(k_1 + k_2 + N) \tag{2}$$

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 rather than 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.

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. This factor, namely $k_3$ in [3], is the keyword's ability to represent the category to which the document belongs. More specifically, $k_3$ can express the positive or negative effect that a keyword's category has on the summarization procedure and is automatically tunable by the core system processes. As long as the text is categorized, we can utilize this factor in order to create a more efficient summary. With the use of $k_3$, the overall weighting equation is depicted in Eq. (3).

$$W_i = \sum (1 + rel(fr(kw_{k,i})))(k_1 + k_2 + N)k_3 \tag{3}$$

### 5.4. Personalization using user feedback

The developed mechanism is based on the continuous feedback from the user. The steps that are followed by the personalization procedure are presented in Algorithm 1. When a new user is registering to the PeRSSonal service, she states the keywords of his/her preference as well as the scores that describe this preference initializing thus his/her profile. This procedure is trivial and can be avoided altogether since the personalization subsystem keeps track of the user's choices and browsing history, and so the user's preferences are updated on each visit. The user's profile consists of two keyword lists: a positive one, where the user-preferred keywords are placed, and a negative one where uninteresting keywords for the user are kept. By using these lists, the summarization procedure can personalize the summaries with exceptional results. Note that a given keyword can only affect a user's profile in a positive or negative manner, but not both ways. We are thus weighting the overall influence of each keyword with a positive or negative outcome.

**Algorithm 1. Personalization algorithm that utilizes user feedback.**

```
Update_profile (a, b, c) {
  Get_articles(a,b)
  for each article {
    if (full article)
        if (time_viewed>Rar_ thr1 && time_viewed< Rar_thr2) {
      Keywords_positive = article's top 5 frequent keywords
      Update_list(Positive, Keywords_positive)}
    else
      //this is a summary
      if(time_viewed > Rsum_thr1 && time_viewed < Rsum_thr2){
        Keywords_positive = article's top 5 frequent keywords
        Update_list(Positive, Keywords_positive)}
    // also recover the negative articles
    Get_articles(c)
    for each article{
      Keywords_negative = select top 5 frequent keywords
      Update_list(Negative, Keywords_negative)
}
Get_article(lists){
//Recovers from the database browsed articles
//and the amount of time spent reading the full article or
//its summary (a,b) Recovers also the negative articles (c)
}
Update_list(list, keywords){
      //either add the keyword to the list or increase its frequency
      for each (keyword in keywords)
      if (keyword not in list[])
        list.add(keywords[keyword])
      else
        list.update_freq(keywords[keyword])
}
```

The profile update procedure, running constantly at every user's visit, takes note of the following aspects: (a) the browsed articles (the ones that the user selected to view), (b) the amount of time a user spends viewing the summary or the full text of a specific article, (c) the articles that the user avoids viewing (either their summary or their full text); the above derives from the simple logical assumptions that follow. A user will most likely spend an amount of time above a certain threshold, $R_{\text{ar\_thr1}}$ or $R_{\text{sum\_thr1}}$, reading an article's full text or its summary respectively, that is of interest to him/her (*factor a*). However, an upper bound, $R_{\text{ar\_thr2}}$ and $R_{\text{sum\_thr2}}$, should be used for these metrics since we donot want the mechanism to mistake forgotten browsed articles for the really interesting ones. The thresholds that are used for $R_{\text{ar\_thr1}}$ and $R_{\text{ar\_thr2}}$ are 30 s and 3 min, respectively, defining thus which article's keywords should be added (or have their weight increased) in the user's positive keywords list. Note that even though an article's length may be varying, we observed that it is quite uncommon for a user to spend over 3 mins reading the same full text article. Consequently we chose the $R_{\text{ar\_thr1}}$ and $R_{\text{ar\_thr2}}$ to be length-independent.

The summary viewing thresholds are calculated in the following way:

$$R_{\text{sum\_thr1}} = R_{\text{ar\_thr1}} * S_{\text{ratio}} \tag{4}$$

$$R_{\text{sum\_thr2}} = R_{\text{ar\_thr2}} * S_{\text{ratio}} \tag{5}$$

Where $S_{\text{ratio}}$ expresses the summarization "compression ratio":

$$S_{\text{ratio}} = \frac{\#words(summary)}{\#words(fulltext)} \tag{6}$$

Moreover, most of the time, a user will select to browse articles of a topic that she finds interesting (*factor b*) as advertised by the article's title or summary. Lastly, a user will probably avoid visiting articles that she finds uninteresting and thus the keywords that represent those articles should be receiving a lessened or negated weight (*factor c*).

From the above factors, the personalization algorithm keeps track of the keywords that the user has expressed preference to and thus, the articles (containing these keywords) that she is likely willing to read in the future. The parameter that depicts the user's preference for a keyword according to the aforementioned factors (a–c) is $U_{wi}$ and is based on the relative frequency that the keyword has on the list, a frequency that is constantly modified by the user's choices. $U_{\mathrm{wi}}$ of the $k$th keyword of sentence $i$, derives from the following equation:

$$U_{wi} = rel(fr(kwi)) * (1 + T_{kwi}) \tag{7}$$

where $T_{kwi}$ is the normalized total time spent on the specific keyword $k$ (of sentence $i$) if it belongs to the positive list; however if the keyword is in the negative list, $T_{kwi}$ is set to 0 since no time is actually spent on these keywords by the user. Furthermore, we expect that when the user profile reaches its steady state, the mean times of the keywords preferences will be correct, hence depicting the overall user preferences.

The overall personalization factor for each keyword $i$, named $k_4$, is:

$$k_{4_i}s = B * U_{wi} \tag{8}$$

where for the parameter $B$: if the keyword belongs to the positive keyword list, then $B > 1$; whereas, if the keyword belongs to the negative keyword list, then $B < -1$. Of course the norm of the $B$ parameter can take any large or short value that we desire, thus increasing or decreasing at will the effect that personalization and dynamic profile generation have on the sentence weighting procedure. From the previous, $|B| > 1$ and thus $k_4$ can be positive, negative or zero in the special condition where no information about the user's preference of the specific keyword is provided.

We could depict the overall user profile as a vector in a multi-dimensional space (sized as per the total keywords) consisting of all the $k_4$ parameters for each keyword. Each user profile has a point, direction and magnitude in the keywords-space. The direction pinpoints the keywords that do actually have an effect on the user's profile, while the magnitude expresses the measure of this effect. This vector is constantly changing as keywords' profile is modified by the user choices. Fig. 2 gives a graphical representation of two user profiles as expressed by their preference for three keywords.

From the previous, the overall weighting formula (3) (with the personalization factor) of the $k$th keyword of sentence $i$, becomes as follows:

$$W_i = \sum (1 + rel(fr(kw_{k,i})))(k_1 + k_2 + N)k_3k_4 \tag{9}$$

### 5.5. Client-side application and content delivery

According to the personalization features of the mechanism that were described earlier, the server transmits the responses to the client using the XML communication protocol derived from the existing XSD schemas. Following the delivery of the content, a number of features are available to the user. Users can browse through the four basic information "channel" tabs: recent/latest, user-preferred, most read and featured, selecting the articles that they are mostly interested in viewing. The summaries of the selected articles are loaded and sent to the application screen. Concurrently, information about the

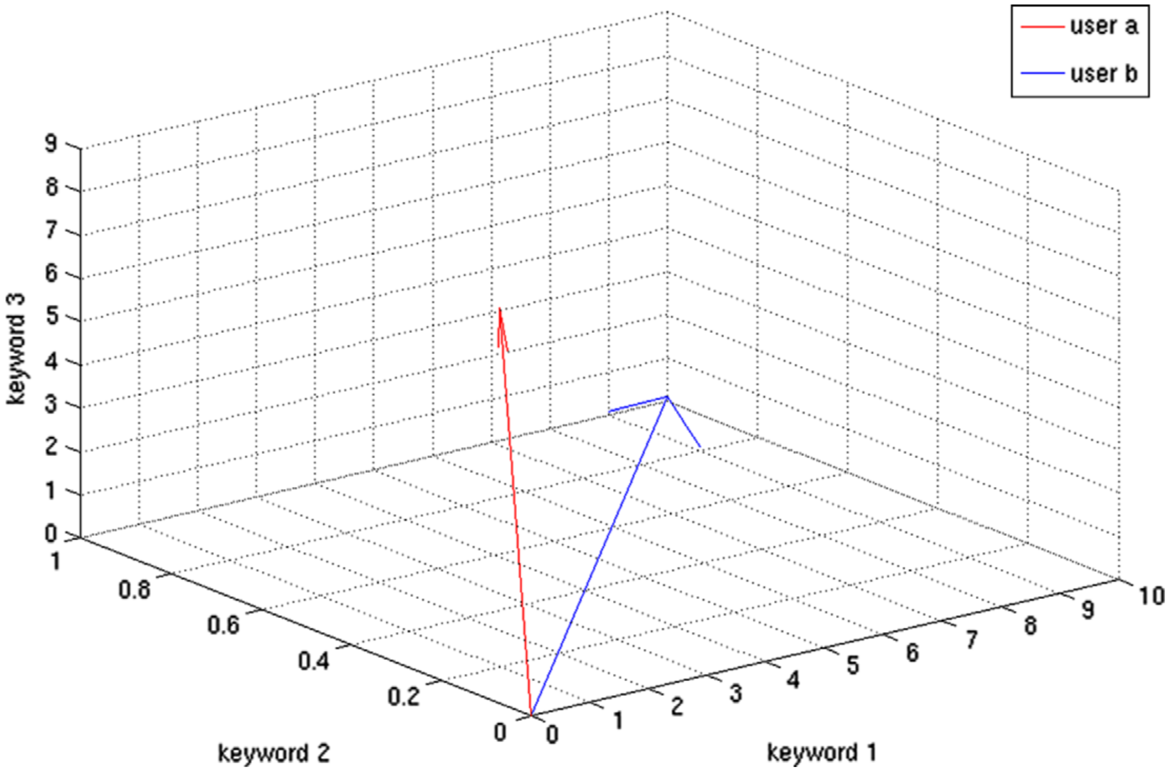


**Fig. 2.** Vector Visualization of the user's preferences.

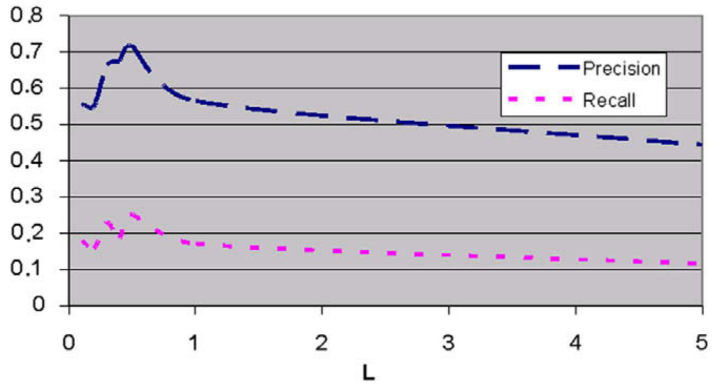


**Fig. 3.** Precision/recall results for summarization of news articles tuning the *L* value.

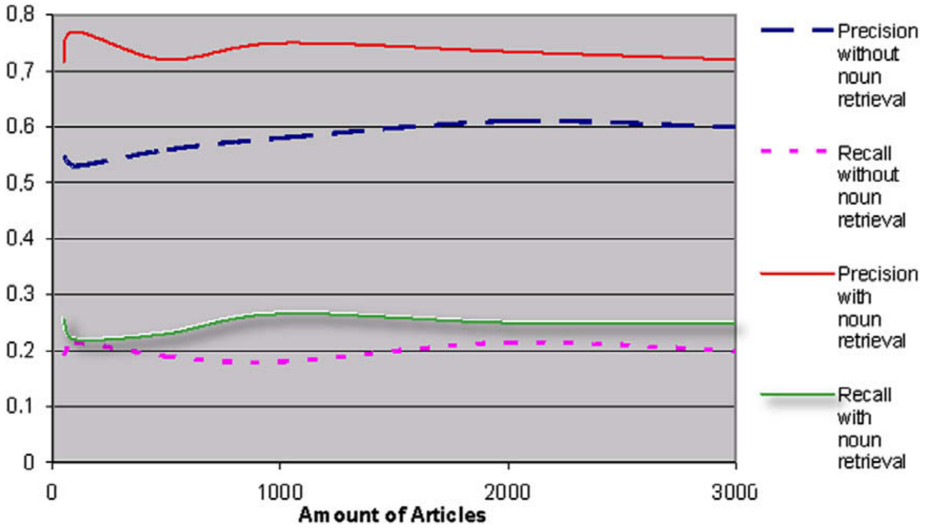


**Fig. 4.** Precision and recall results for the PeRSSonal's summarization procedure when noun retrieval information is utilized and when it is not.

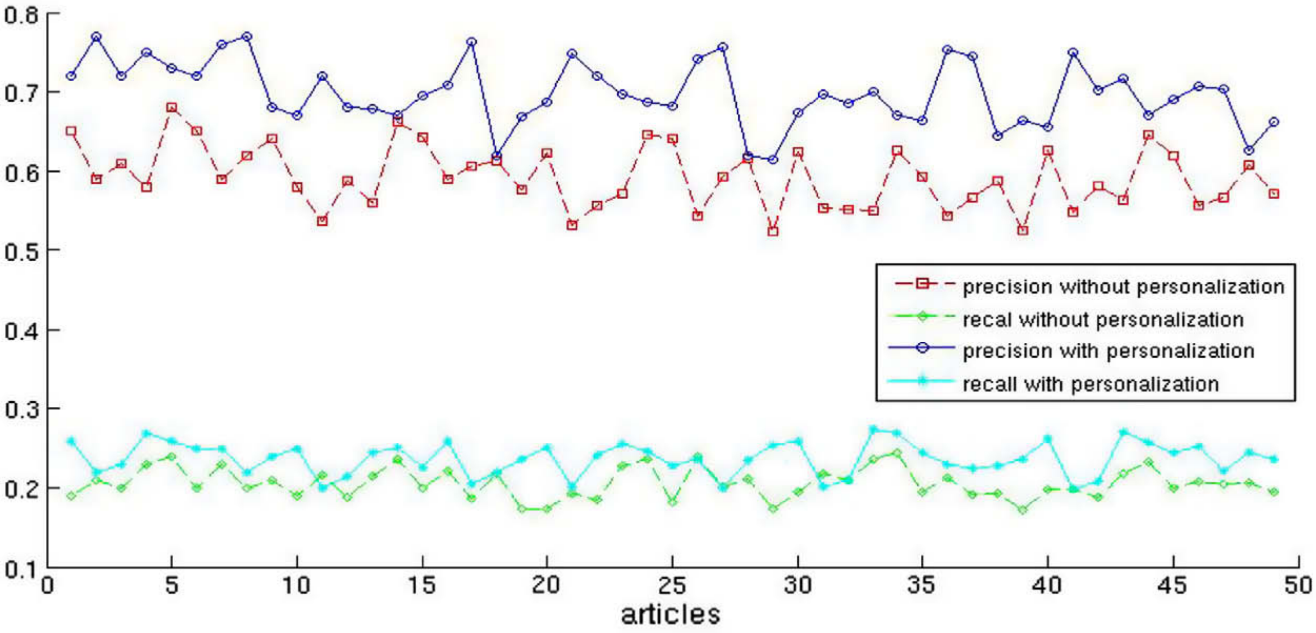
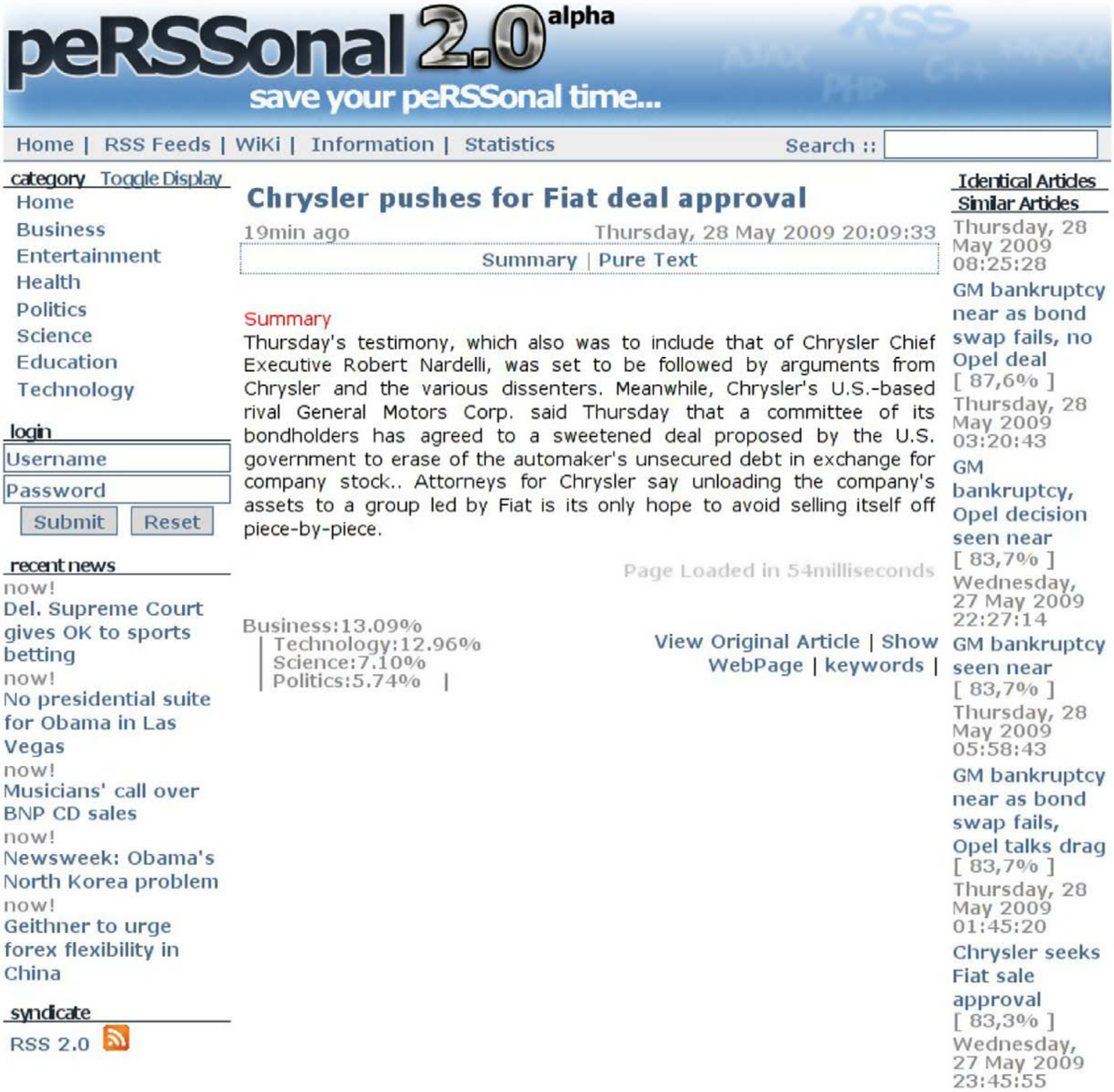


**Fig. 5.** Precision and recall Results.

related, identical and similar articles is fetched and suitably depicted. The whole application is fully configurable; working in a modular manner, since the GUI is dynamically created using the existing XSD schema. For example, should a future version of PeRSSonal include a new channel, the client application will recognize the addition offering the user the choice to place the new module according to his/her desire.

It is also important to note that information is loaded in a multithreaded manner by the client. Having available information about the user's profile and browsing habits, we are able to pre-fetch articles, summaries and relativity information before the user actually asks for it. This implementation offers a significant boost to our application over the web interface of PeRSSonal, since it makes it more responsive and usable. For instance, consider the following scenario: a user's profile contains many keywords from the politics category with high preference (as explained in Section 5.4) and from the fetched "latest" channel's articles, some of them are related with high similarity to this category – more precisely, the similarity check is done upon the keywords, so an article belongs with high similarity to a category when the majority of its keywords belong to this category, see [3] (categorization subsystem) for more details – the article's summary and relativity information are automatically loaded and stored for future use.



Chrysler pushes for Fiat deal approval

19min ago Thursday, 28 May 2009 20:09:33

Summary

Thursday's testimony, which also was to include that of Chrysler Chief Executive Robert Nardelli, was set to be followed by arguments from Chrysler and the various dissenters. Meanwhile, Chrysler's U.S.-based rival General Motors Corp. said Thursday that a committee of its bondholders has agreed to a sweetened deal proposed by the U.S. government to erase of the automaker's unsecured debt in exchange for company stock.. Attorneys for Chrysler say unloading the company's assets to a group led by Fiat is its only hope to avoid selling itself off piece-by-piece.

Page Loaded in 54milliseconds

Business:13.09%
Technology:12.96%
Science:7.10%
Politics:5.74%

now! Del. Supreme Court gives OK to sports betting
now! No presidential suite for Obama in Las Vegas
now! Musicians' call over BNP CD sales
now! Newsweek: Obama's North Korea problem
now! Geithner to urge forex flexibility in China

Thursday, 28 May 2009 08:25:28
GM bankruptcy near as bond swap fails, no Opel deal
[ 87,6% ]

Thursday, 28 May 2009 03:20:43
GM bankruptcy, Opel decision seen near
[ 83,7% ]

Wednesday, 27 May 2009 22:27:14
GM bankruptcy seen near
[ 83,7% ]

Thursday, 28 May 2009 05:58:43
GM bankruptcy near as bond swap fails, Opel talks drag
[ 83,7% ]

Thursday, 28 May 2009 01:45:20
Chrysler seeks Fiat sale approval
[ 83,3% ]

Wednesday, 27 May 2009 23:45:55

**Fig. 6.** The system's Web interface.

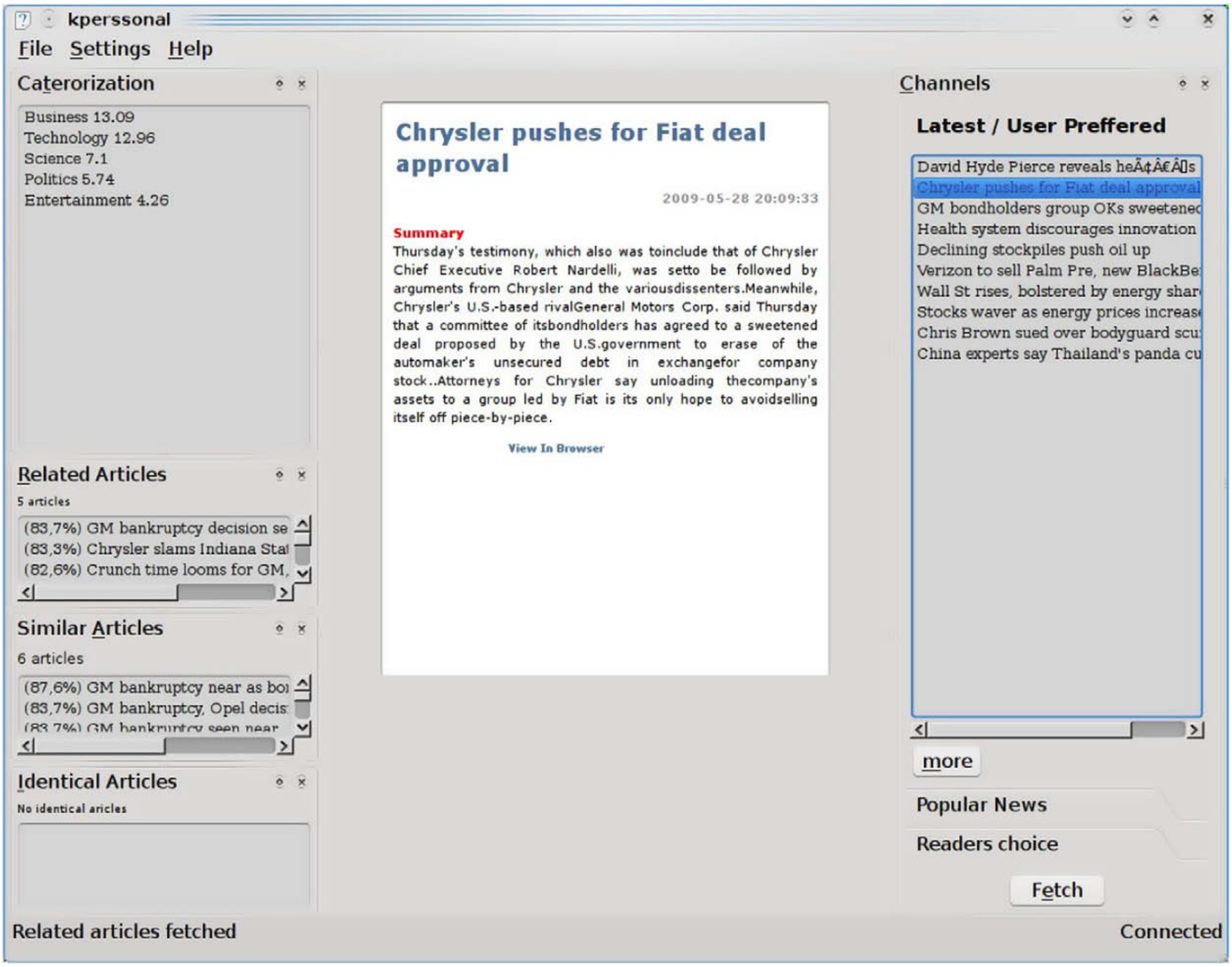
## 6. Experimental procedure and results

The current section presents the experimental evaluation of PeRSSonal with the application of noun retrieval techniques as well as the personalization algorithm that was presented earlier. Moreover, we evaluate the desktop application (presentation subsystem) in order to detect its acceptance level by the system's users. For our experimentations we blocked any unrelated access to the system's web interface so as to be sure that the results observed are accurate.

### 6.1. Noun retrieval effects in summarization

In order to evaluate the summarization performance of PeRSSonal, with the appliance of noun retrieval techniques, we conducted two sets of experiments. Firstly, we tried to determine the best possible value for the $L$ parameter of Eq. (1). Furthermore, we tried to evaluate the effect of the appliance of the noun retrieval algorithm explained earlier, to the overall system performance using classic IR measures. For conducting the experiments we used a corpus of 3000 news articles that where obtained from major news articles portals from around the Web (namely: CNN, BBC, Reuters, MSNBC, ABCNews, Washington Post and Guardian). The articles belonged with high relevance to one of the seven major categories of the system, and this information was used as explained in Section 5 (pre-categorized articles) in order for the summarization procedure to produce the best possible summary. The categories that the system served for our experimentation were: business, entertainment, health, politics science, sports and education. We used a same article amount for each of the system's categories so as not to bias any specific one.

As reported earlier, the parameter $L$ is deployed for controlling the effect that noun retrieval has on the article summarization procedure. We conducted experiments tuning $L$ in order to decide on its best value as far as news articles, which is the case of PeRSSonal, are concerned. The various results are presented in Fig. 3.



Business 13.09
Technology 12.96
Science 7.1
Politics 5.74
Entertainment 4.26

5 articles

(83,7%) GM bankruptcy decision se
(83,3%) Chrysler slams Indiana Stat
(82,6%) Crunch time looms for GM,

6 articles

(87,6%) GM bankruptcy near as boı
(83,7%) GM bankruptcy, Opel decis
(83,7%) GM bankruptcy seen near

No identical aricles

Chrysler pushes for Fiat deal approval

2009-05-28 20:09:33

Summary

Thursday's testimony, which also was toinclude that of Chrysler Chief Executive Robert Nardelli, was setto be followed by arguments from Chrysler and the variousdissenters.Meanwhile, Chrysler's U.S.-based rivalGeneral Motors Corp. said Thursday that a committee of itsbondholders has agreed to a sweetened deal proposed by the U.S.government to erase of the automaker's unsecured debt in exchangefor company stock..Attorneys for Chrysler say unloading thecompany's assets to a group led by Fiat is its only hope to avoidselling itself off piece-by-piece.

David Hyde Pierce reveals heÃ¢Â€Â▯s
Chrysler pushes for Fiat deal approval
GM bondholders group OKs sweetened
Health system discourages innovation
Declining stockpiles push oil up
Verizon to sell Palm Pre, new BlackBe
Wall St rises, bolstered by energy shar
Stocks waver as energy prices increase
Chris Brown sued over bodyguard scu
China experts say Thailand's panda cu

Related articles fetched

Connected

**Fig. 7.** The Desktop application developed for PeRSSonal.

In the graph of Fig. 3 it is clearly depicted that a value between 0.5 and 0.7 for $L$ is best fitted. Values for $L$ over 1 seem to attenuate both the precision and the recall of the summarization procedure compared to the $L = 0$ case, i.e. when noun retrieval information is not used. This intuitively means that, when sentences that contain mostly nouns are kept at the summarization procedure (i.e. large $L$ values), excluding the rest of the sentences, the effectiveness of the summarization procedure slightly deteriorates. However, finding a golden section for the $L$ parameter, which is dependable on the target texts, can enhance the summarization efficiency significantly. The value of $L$ that we observed to give high precision/recall results was from 0.5 to 0.7, depending on the category of the texts. Since this span for the $L$ parameter is relatively low, it is normal to suppose that it is due to statistic anomalies of the used text set. After all, there seems to be no logical explanation as to why some category should benefit more that someone else when weighting its nouns with a higher or lower $L$ parameter. On the other hand, we are skeptical on the assumption that using similar $L$ values for any kind of input texts (such as e-mails or published papers) would give similar results. The same applies with the case of more article texts since the observed result was with an experimental training set of 3000 texts; however we believe that the observed $L$ values can definitely give a good estimation even for a larger article set.

The aforementioned improvement to the summarization results is also obvious at the following graph, where precision and recall results are depicted (using an $L$ value of 0.6) when summarization proceeds with and without the noun retrieval information.

From Fig. 4 it is concluded that the noun retrieval information can give a notable precision boost to the resulting summaries compared to the case where noun retrieval information is not utilized; in other words, the resulting summaries are more precise. As far as recall is concerned, the improvement is small, yet significant, taking into account the fact that a text's summary represents a layer of abstraction, notably a low recall representation of the original text's information; expecting thus high recall improvements would not be wise.

### *6.2. Evaluation of the personalization and presentation subsystem*

In order to evaluate the possible enhancement of the proposed personalization mechanism on the system's procedures, and more specifically, on the summarization subsystem, we conducted a set of experiments. We also experimented in order to determine whether the client-side desktop application is actually a nice equivalent to the web interface.

For our summarization evaluation approach, we used classic precision–recall metrics and 15 university students. We first asked our test users to register to the system and use it (through its web interface and the desktop application) for one month's period so that the personalization algorithm fully adapts their profile to their preferences. Afterwards, we provided them 50 full text articles that were matching their created profiles and we asked them to rate some sentences of these arti-

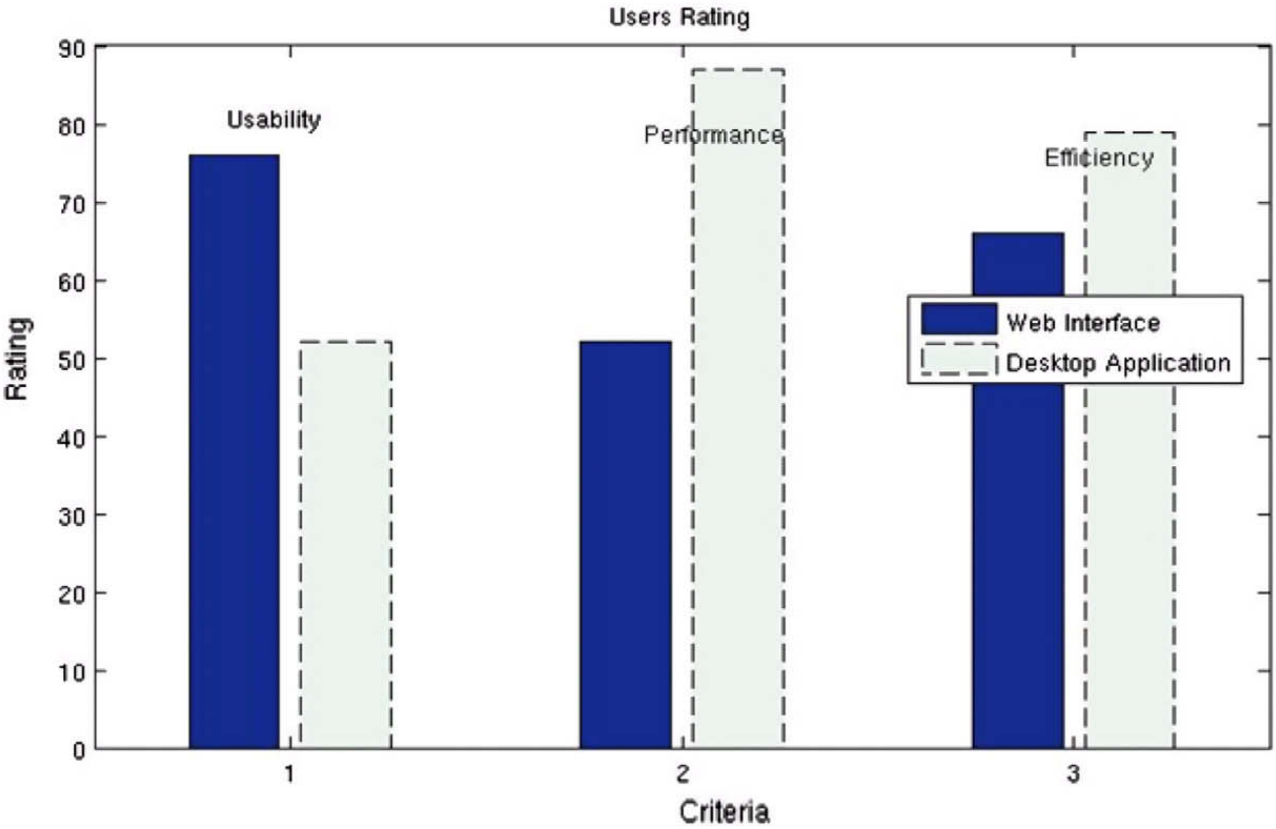


**Fig. 8.** Users evaluation of the presentation subsystem.

cles for a suitable summary of the article. We also produced the personalized as well as the "generic" summaries of these articles (without utilizing the personalization features explained in this article) and compared them with the users' choices in terms of precision and recall. The results are presented in Fig. 5. They are depicting the rates without the personalization factor which are based on the generic summaries that the system generates for the given articles. Extracted summaries are then compared with the sentences selected by the users and precision as well as recall metrics are evaluated.

From Fig. 5 it is deducted that the appliance of our new personalization scheme has provided a significant benefit to the overall summarization performance of the mechanism as far as precision and recall metrics are concerned. We measured this increase to be around 17% for precision and 14% for recall. It is important however to note that the statistics are based on the user choices and are subjectively biased by nature. On the other hand though, there are no real objective criteria for the extraction of a summary from a text and it is this "bias" that the proposed personalization mechanism is trying to estimate.

As already explained, the system features two ways for presenting information to the user: the Web interface and the Desktop application. Figs. 6 and 7 give a depiction of both. Note that the presented amount of actual news-related information is equal in both of the cases.

Moving to the next set of evaluation, we asked our users to utilize both the web interface and a beta version of the client-side application of PeRSSonal for a period of 30 days. We then asked them to rate both of the presentation systems in terms of: (a) usability (is the system serving enough and good summaries? Is everything reachable within a few mouse-clicks? Is the displayed information easily understood?) and user-friendliness, (b) performance and interactivity (are response times good?) and (c) efficiency and briefing in content representation. The rating was done with a scale from one to ten, with ten being the best.

The results that are presented in Fig. 8, express the user's overall preference in favor of the desktop client-side application for the presentation subsystem of PeRSSonal. It is clear though that some users thought low of the desktop application in terms of its usability since they often got puzzled on its use, especially as far as discovering all the features that it provides is concerned. This is however expected since (a) to our knowledge this is the first desktop application that focuses on such information retrieval and personalization tasks, (b) taking into consideration its compact representation of quite a big amount of information.

Furthermore, as far as performance and interactivity are concerned, the desktop application outruns the web interface. This originates from the caching and pre-fetching techniques that the desktop application makes use of. Lastly, efficiency and briefing in content representation, a key target of the PeRSSonal system is an aggregation of the previous factors and shows that despite the fact that the client-side application is still under development, the novel features that it provides are considered overall useful by the users. Even though the above results depend on highly subjective user choices, they give us a useful feedback for as far as the current status of the developing application is concerned, as well as the future directions that we should look into.

## 7. Conclusions and future work

In this article we explored the effects that noun retrieval techniques, based on POS tagging, can have on information retrieval mechanisms and summarization in specific. We have also outlined the personalization algorithm that our system utilizes for presenting pre-categorized and summarized articles to the user. Furthermore, we presented the communication channel that PeRSSonal uses for delivering content to the end users, as well as the desktop application, which is capable of exchanging information with our mature system. Through the proposed framework that is utilized in an existing system, PeRSSonal, we are able to improve the summarization procedure by simple modifications to our keyword extraction algorithm. Our personalization approach is mainly content-based with some collaborative filtering features adopting over time to the continuously changing user profile.

We conducted experimental procedure in order to evaluate the overall improvement of PeRSSonal's summarization capabilities with the appliance of the new personalization algorithm, as well as the new keyword extraction capabilities. We used real system users and even though the evaluation of a summarization system is a difficult and subjective task, we discovered a significant amendment. Moreover, we evaluated the developing client-side desktop application and explored the significance of developing such a communication infrastructure from the point of the end user. The results are encouraging and express the users' need for efficiency and interactivity from an application that is targeted not as a replacement, but as a complement of the Web 2.0 technologies that are utilized by the system. The efficiency improvements concerning the applied noun identification technique are small, yet significant, considering the fact that summarization is a difficult, mostly subjective procedure and that objective criteria of efficiency are difficult to appoint.

Having incorporated noun retrieval techniques, as far as the core procedures of the system are concerned, we intend to incorporate multilingual support, as well as support for multimedia and improved caching features. The addition of such features to PeRSSonal will require a basic redesign of the main parts that constitute the system. Furthermore, we are focusing on a stable version of our desktop application and on a wider evaluation of PeRSSonal. Also, we consider a wider evaluation of the improvements that the applied noun retrieval technique has on both the summarization and the categorization procedure, as well as evaluation on how the system behaves with regard to different paces of user preference changes.

## References

[1] E. Banos, I. Katakis, N. Bassiliades, G. Tsoumakas, PersoNews: a personalized news reader enhanced by machine learning and semantic filtering, Lecture Notes in Computer Science 4275 (2006) 975.

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

[3] C. Bouras, V. Poulopoulos, V. Tsogkas, PeRSSonal's core functionality evaluation: enhancing text labeling through personalized summaries, Data and Knowledge Engineering Journal 64 (1) (2008) 330–345.

[4] D. Cutting, J. Kupiec, J. Pedersen, P. Sibun, A practical part-of speech tagger, in: Proceedings of the 3rd Conference on Applied Natural Processing, 1992, pp. 133–140.

[5] Document Object Model (DOM) - W3C, http://www.w3.org/DOM/.

[6] Extensible Markup Language (XML) - W3C, http://www.w3.org/TR/1998/REC-xml-19980210.

[7] P. Ferragina, A. Gulli, A personalized search engine based on web-snippet hierarchical clustering, in: Proceedings of WWW Conference Vol. 38 (2) 2005, pp. 189–225.

[8] E. Gabrilovich, S. Dumais, E. Horvitz, Newsjunkie: providing personalized newsfeeds via analysis of information novelty, in: Proceedings of the 13th international conference on WWW, ACM, 2004, pp. 482–490.

[9] J. Gimenez, L. Marquez, SVMTool: A general POS tagger generator based on Support Vector Machines, in: Proceedings of the 4th International Conference on Language Resources and Evaluation, 2004, pp. 43–46.

[10] J. Goldstein, et al., Summarizing Text Documents: sentence selection and evaluation metrics, in: Proceedings of ACM SIGIR Conference, 1999.

[11] P.J. Hayes, et al., A news story categorization system, in: Proceedings of the second Conference on Applied Natural Language Processing, 1988, pp. 9–17.

[12] F. Karlson, A. Voutilainen, J. Heikkila, A. Anttila, Book: Constraint Grammar, A Language-Independent System for Parsing Unrestricted Text, Mouton de Gruyter, 1995.

[13] H. Luhn, The automatic creation of literature abstracts, Presented at IRE National Convention, New York, March 24, 1958.

[14] K. McKeown, R. Barzilay, D. Evans, Columbia multi-document summarization: approach and evaluation, in: Proceedings of the Workshop on Text Summarization, ACM SIGIR Conference, 2001.

[15] O. Nasraoui, World wide web personalization, in: J. Wang (Ed.), Invited chapter in Encyclopedia of Data Mining and Data Warehousing, Springer, 2005.

[16] PeRSSonal official website, http://perssonal.cti.gr.

[17] D. Roth, D. Zelenko, Part-of-speech tagging using a network of linear separators, in: Proceedings of the 36th Annual Meeting of the ACL – Coling, Mondreal, Canada, 1998, pp. 1136–1142.

[18] Simple API for XML (SAX), http://www.saxproject.org/.

[19] C. Wongchokprasitti, P. Brusilovsky, NewsMe: a case study for adaptive news systems with open user Model, Proceedings of the Third International Conference on Autonomic and Autonomous Systems, IEEE Computer Society, 2007. p. 69.

[20] M. Wasson, Using leading text for news summaries: evaluation results and implications for commercial summarization applications, in: Proceedings of ICCL, 1998, pp. 1364–1368.

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

**Christos Bouras** obtained his Diploma and PhD from the Department Of Computer Engineering and Informatics of Patras University (Greece). He is currently a 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 300 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 Tsogkas** is a PhD candidate at the Computer Science and Engineering Department of Patras University (Greece). He obtained his diploma as well as his Master's degree in Computer Science from the above department. His basic fields of interest are: source code management, PHP, MySQL, HTML programming, C/C++ and Java programming. He has published over 10 papers in various well-known refereed conferences and journals. 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 and Clustering.