Iglesias, J.A., Tiemblo, A. Ledezma, A. Sanchís, A. (2016). Web news mining in an evolving framework. *Information Fusion*, 28, pp. 90-98.

DOI: 10.1016/j.inffus.2015.07.004

Web News Mining in an Evolving Framework

José Antonio Iglesias[a], Alexandra Tiemblo[a], Agapito Ledezma[a], Araceli Sanchis[a]

[a] *University Carlos III of Madrid, 28911, Leganes, Madrid, Spain*

Abstract

Online news has become one of the major channels for Internet users to get news. News websites are daily overwhelmed with plenty of news articles. Huge amounts of online news articles are generated and updated everyday, and the processing and analysis of this large corpus of data is an important challenge. This challenge needs to be tackled by using big data techniques which process large volume of data within limited run times. Also, since we are heading into a social-media data explosion, techniques such as text mining or social network analysis need to be seriously taken into consideration.

In this work we focus on one of the most common daily activities: web news reading. News websites produce thousands of articles covering a wide spectrum of topics or categories which can be considered as a big data problem. In order to extract useful information, these news articles need to be processed by using big data techniques. In this context, we present an approach for classifying huge amounts of different news articles into various categories (topic areas) based on the text content of the articles. Since these categories are constantly updated with new articles, our approach is based on Evolving Fuzzy Systems (*EFS*). The *EFS* can update in real time the model that describes a category according to the changes in the content of the corresponding articles. The novelty of the proposed system relies in the treatment of the web news articles to be used by these systems and the implementation and adjustment of them for this task. Our proposal not only classifies news articles, but it also creates human interpretable models of the different categories. This approach has been successfully tested using real on-line news.

Keywords: Big Data, Web News Mining, Evolving Fuzzy Systems

Email addresses: jiglesia@inf.uc3m.es (José Antonio Iglesias), 100073062@alumnos.uc3m.es (Alexandra Tiemblo), ledezma@inf.uc3m.es (Agapito Ledezma), masm@inf.uc3m.es (Araceli Sanchis)

1. Introduction

Modern society generates huge amounts of information every day, especially in digital format, which obstruct the storage and further processing and analysis. Big data can be defined as a scale of data set that goes beyond existing database management tool capabilities of data collection, storage, management, and analysis capabilities [1]. Although the most common trait of *big data* is Volume, it is typically defined by three *Vs* (Volume, Variety and Velocity).

In addition, big data can be classified taking into account the data type:

1. Structured (data are organized into a predefined data schema)
2. Semi-structured (data does not require a schema definition but the data includes metadata)
3. Unstructured (data are stored in an unstructured form without any defined data schema)

There are many different applications in which big data techniques are applicable: data mining, predictive analytics, geoanalysis, natural language processing and pattern recognition. Also, we are heading into a social-media data explosion. In this connection, web-based applications encounter big data frequently, such as *social computing*, Internet text and documents or Internet search indexing. For this reason, there are some techniques which need to be seriously taken into account such as social network analysis and text mining.

In particular, an important application of the text mining which could be also related with the social big data is: web news mining. Since news websites are daily overwhelmed with plenty of news articles, an important part of the huge amounts of *new* information produced each day is generated by the on-line newspapers. For this reason, automatic systems which can treat, analyze and classify web news articles are essential not only for those systems which manage web news articles but also for user recommendation tasks.

According to the Statistical Report on Internet Development released by China Internet Network Information Center (CNNIC) in July 2014 [2], the number of online news users in China had reached 503 million by the end of January 2014 (a growth of 98.60 millions from June 2012), and the utilization ratio of online news was 79.6%. In this report, the *online news* is the third most used network application by Internet users. The first application in this ranking is the *Instant messaging* (89,3%) and the second one is the *Search engine* (80,3%). However, news is the most frequently searched content by the Internet users using both computers and mobile phones. In addition, this report suggests that online news has become one

of the major channels for Internet users to get news and its utilization ratio has remaining high due to the following reasons: 1) in the era of mobile Internet, it is one of major activities of Internet users to read news in their fragmented time, 2) Internet users can get news through more channels, and 3) all news media vied with each other to make inroads into the mobile Internet.

In the era of big data and because of this explosion of information from news websites, extracting knowledge from news articles becomes an interesting challenge. To that end, we need text mining techniques which can extract relevant information from this kind of unstructured type text data. In addition, online news is a special type of public information mainly because there are many news sources and the update of the news is very fast. News mining tools, techniques, and algorithms are strongly emerging during these times. There are many techniques which help to analyze the overflow of information and extract value knowledge from on-line news sources. However, since this information is continuously growing and changing, these techniques have to skim and search for information much more than they had to do in the past.

During the last years, there have been many approaches related with classification, clustering, categorization and summarization of news articles [3, 4, 5, 6, 7, 8]. If we consider those approaches which classify news into predefined categories, all of them use a statistic classifier over time. However, the news articles of the different categories change constantly and these changes should be considered in the model of the classifier. For this reason, we propose an approach in which the categories are not predefined but they are updated in an evolving manner according the new news articles and categories obtained. This aspect makes our approach an ideal alternative in this environment.

The presented approach is based on Evolving Fuzzy Systems (EFS) [9] which allows not only update the structure and parameters of an evolving classifier but also cope with huge amounts of web news and process data in on-line and real time - which is essential in this (web) environment. EFS approaches have been successfully applied in many other different areas [10, 11, 12, 13, 14, 15] and for big data problems [16].

The remainder of the paper is organized as follows: Section 2 describes existing researches and approaches related with the area of big data and web news mining. Sections 3, 4 and 5 describe our proposed (evolving) approach for the classification of web news. Section 6 presents the results and analysis of the evaluation of our approach. Finally, section 7 draws the conclusions and future work guidelines.

2. Background and Related Work

Many different scientific fields have become highly data-driven with the development of computer science. Social computing [17], astronomy [18] or bioinformatics [19] are some examples of these fields.

Big data uses different techniques to efficiently process large volume of data within limited run times. Because of the most common trait of big data is Volume, the most important challenge is scalability when we deal with the big data analysis tasks. In this sense, incremental algorithms have good scalability property [20, 21]. If we focus on the disciplines of data mining and machine learning, we should consider that big data mining is more challenging compared with traditional data mining algorithms [22].

However, in the big data era, we have to consider that the most common format of information storage is text such as web pages, emails, documents or social media. For this reason, text analysis or text mining is a powerful technique at that time. The term *text mining* or *Knowledge Discovery from Text* (KDT) was mentioned for the first time in 1995 by Feldman et al. [23]. They propose to structure the text documents by means of information extraction, text categorization, or applying NLP techniques as pre-processing step before performing any kind of KDTs.

Text mining, also known as text data mining[24], can be defined as the analysis of semi-structured or unstructured text data. As the text is in unstructured form, it is quite difficult to deal with it. In fact, text mining is a much more complex task than data mining [25] as it involves dealing with text data which are inherently unstructured and fuzzy. Thus, the goal of the text mining is to turn text information into numbers so that data mining algorithms can be applied. It arose from the related fields of data mining, artificial intelligence, statistics, databases, library science, and linguistics. As it is detailed in [3], since text mining is a multidisciplinary field, this term has been used to describe different applications such as text categorization [26, 27], prediction [28, 29], text clustering [30, 31], association discovery [32, 33] and finding patterns in text databases [34].

In the text mining area, Twitter is considered as a rich source of information for text analysis. In [35], the authors find similarities between tweets before the World Cup started. The high-value social audience from Twitter is identified through text-mining methods [36]. In this case, the Twitter content of an account owner and its list of followers are analyzed. A survey on text mining and sentiment analysis for unstructured web data is presented in [37]. Mathioudakis et al. [38] propose *TwitterMonitor*, a system which detects topic trends in real time

and provides meaningful analytics that synthesize an accurate description of each topic. Kim et al. propose in [39] a spatio-temporal trend detection and related keyword recommendation scheme for tweets called *TwitterTrends*. These scheme can identify keyworkds and recommmend related keywords at a given location and time.

Other application of the text mining is: Web news mining. This term describes the analysis of web news and is a special type of public information which has special characteristics [40]. The existence of numerous reliable news sources and fast news updates are two important differences. For this reason, new approaches, technologies and tools need to be developed in order to achieve the different goals proposed in this area.

During the last years, there have been many approaches related with web news mining and news exploration systems. In [6], the authors describe the use of data mining techniques to analyze web news. It is concluded from that study that web mews mining at the terms level serves as a powerful technique to manage knowledge encapsulated in large web news collection. As in our approach, the authors analyze web news by using text mining. However, that research only implements the process of terms extraction from the web news. Our approach, not only analyzes web news but also classifies them in a specific topic.

In [41] the authors propose a flexible topic-driven framework for news exploration. It performs news mining at the topic level and presents news information with topics, entities and relations derived from the news data. Also, the authors consider that in order to facilitate an in-depth analysis of the news it is necessary to extract structured information (ideally, identifying *who, what, whom, when, where* and *why* [42]). In [43], it is presented an endeavor aiming at construction of a real-time event extraction system for border security-related intelligence gathering from online news. In [44] a quantitative method that identifies weak signal topics by exploiting keyword-based text mining is presented. This method is illustrated using web news articles related to solar cells.

Because the amount of web news is huge, there is also a need for approaches which can help people to extract the most important information very quickly. For this reason, automatic news summarization has been an active area of research for several decades. Malhotra et al [45] propose a technique which is keyword based extractive summarization. In that case, different features (such as thematic terms, named entity or title terms) are identified and used to score sentences. In a recent research, Chowdhury et al. [46] explain that since each of the news carries information, news pertaining to a specific company can give us a perception about the organizations policies, growth and performance. In this sense, all news together

can build an overall sense about a particular company. For this reason, the authors propose an interesting prediction model which shows the current sentiment of a company calculated from relevant news headlines.

In the area of news summarization, there are also several studies in which hybrid systems based on users web history are proposed. In the research done by Liu et al. [47], a personalized news recommendation system in *Google News* is developed. Their system builds profiles of users's news interests based on their past click behavior. In [8] Kim et al. propose a news summarization scheme based on social network services which generate detailed information about trending issues in an effective manner. Morales et al. [48] propose a methodology for recommending interesting news to users by exploiting the information in their twitter persona. In this approach, it is analyzed the relevance between users and news articles using a mix of signals drawn from the news stream and from twitter (the profile of the social neighborhood of the users, the content of their own tweet stream, and topic popularity in the news and in the whole twitter-land).

Automated summarization methods are defined as *language-independent* if they are not based on any language specific knowledge. These methods can be used for multilingual summarization. This term was defined by Mani [49] as processing several languages, with summary in the same language as input. In the period since 2004, an interest in multilingual and multi-document summarization has risen. Evans et al. [50] proposed in 2004 a multilingual version of a summarization system which address the problem of user access to browsing news from multiple languages from multiple sites on the internet. The system automatically collects, organizes, and summarizes news in multiple source languages. Also, the user can browse news topics with English summaries, and compare perspectives from different countries on the topics. A technique which fusion sentences by summarizing news in multiple documents is presented by Barzilay et al. [4]. In [51] the authors introduce an approach to multilingual single-document extractive summarization where summarization is considered as an optimization or a search problem which is solve by using genetic algorithms. Recently, Kabadjov et at. [7] present a generic approach for summarizing clusters of multilingual news articles by using statistical techniques and multilingual tools.

In conclusion, in the era of web pages, emails and social media; text mining presents new opportunities and challenges. In particular, news websites produce huge amounts of articles which can be considered as a big data problem. For this reason, web news mining has become an attractive research area which needs to be considered by using big data techniques. In this paper, we present an evolving approach for classifying web news articles which can be used on-line, which is

essential in this (web) environment. To the best of our knowledge, this is the first approach which treat the web news mining in an evolving manner.

3. Our Approach: General Structure

As we have already mentioned, the goal of this research is the development of an approach to classify different news articles (from the web) into various topic areas (categories) based on the text content of the articles. The cornerstone of the proposed approach is the use of a classifier whose structure and parameters are updated according to the changes in the content of the news articles.

Figure 1 shows the structure of our approach, which consists of two well differentiated phases (or modules):

1. *Term extraction:* creates a set of (relevant) terms per article (this module is explained in Section 4)
2. *Evolving Classification:* not only creates and constantly updates the corresponding Evolving Fuzzy Rules from the obtained articles, but also classifies a new news article into the categories previously considered (in Section 5 this module is detailed).

The extraction and analysis of web news articles are done one by one and this process can start *from scratch*, with no previous information about the number of categories or news articles collected. It is important to highlight that all the approaches that we have seen so far classify news into predefined categories using a predefined and static classifier over time. As far as we know, this is the first approach which not only collects, analyzes and extracts relevant terms from different web news, but also classifies web news in an *evolving* manner.

The proposed classifier also can cope with huge amounts of web news articles and process them in real time. This aspect is essential because of the characteristic of this big data problem in the web environment. Although our proposal is focused on the web news mining, it can be used in other web-based applications, like navigation or social networks, that produce big data in real time.

The next two sections explain in detail how the proposed approach works.

4. Our Approach: Term extraction

This module is responsible for extracting news articles of different topic areas from the web, and then summarizes each article with a set of terms in which each term has its corresponding relevance value.

Although we can use any source for the articles collection, in this research we have collected them from the New York Times (*NYT*) online newspaper [1]. For such a task, we have used the *NYT API* [2] called *Article Search API v2* with which we can search *NYT* articles from September 18, 1851 to today, retrieving headlines, abstracts, lead paragraphs, links to associated multimedia and other article metadata. Using this API, we can obtain search NYT articles based on their category (using the search filter *new_desk*) and their date of publication (*begin_date* and *end_date*). By using these search filters, we obtain a JSON response with several parameters. In this research, we will use only the content of the parameter (*lead_paragraph*) which contains the first paragraph of the web news as it was published in the web. However, other parameters (such as *section_name*, *web_url* or *word_count*) could be used if we wanted to analyze other aspects.

In Figure 1we can observe that once a categorized news article (A_k) is obtained, it is considered as a *string* and analyzed by applying the following two steps: *Term Generation* (section 4.1) and *Term Filtering* (section 4.2).

4.1. Term Generation

The most relevant terms of each article are obtained in this step. This task has been done in this research by using the open-source tool *RapidMiner* [52]. *RapidMiner* is one of the most popular tools for data mining and predictive analysis. This tool is used for executing the following steps:

1. Tokenization: This step breaks a stream of text up into phrases, words, symbols, or other meaningful elements called tokens. The result of the tokenization is a sequence of tokens, and its main use is the identification of meaningful keywords.
2. Stopword elimination: The most common words that are unlikely to help text mining such as prepositions, articles, and pro-nouns are considered as stopwords. This step eliminates these words from the text because they are not useful for the text mining applications. Thus, the sequence of tokens is reduced and it helps to improve the system performance.
3. Stemming or lemmatization: This step reduces the words into their stems (also known as base or root). Since the meaning of different words could be the same but their form different, it is necessary to identify each word form using its stem form. There are many stemming algorithms which can

[1] http://www.nytimes.com/

[2] http://api.nytimes.com/svc/search/v2/articlesearch

do this. In this case, we have applied the algorithm used by the Snowball Stemmer [53]. Thus, the terms that we are using are represented by stems rather than by the original words. In this step, the total number of distinct terms in a document is reduced too.

4.2. Term Filtering

The previous module (*term generation*) produces a set of terms associated with each document. However, the relevance of the different terms in the framework of all the news should also be taken into account. For this reason, it is necessary to prune the generated terms based on their frequencies of occurrence throughout the collection. The aim of this term filtering process is to identify those terms which are not of interest in the context of the entire news corpus. Thus, we need to remove not only the terms which do not occur frequently enough but also those which occur in a fairly constant distribution among the news collection.

In order to assign a *relevance value* to each term, we propose an Information Retrieval (IR) approach. In this case, the term relevance with respect to a news article collection is obtained by using one of the most successful and well-tested techniques in IR: *tf-idf* [54] (Term Frequency - Inverse Document Frequency). *Tf-idf* is a metric that in this case determines the relative frequency of words in a specific news article compared to the inverse proportion of that word over the entire news article corpus. This metric provides how relevant a given word is in a news article. Those words which appear in a small group of articles will have higher *tf-idf* value than those words which are very common. Since the calculation of this metric needs the entire news article corpus, it is updated with each new news article. It is proper to remark that although our approach starts *from scratch*, the initial corpus could be created by analyzing an initial set of articles.

The output of this process per news article (Figure 1 shows an example) is the corresponding *tf-idf* value per each term of the corpus. We need to take into account that this module is also responsible for the (constantly) update of the corpus.

5. Our Approach: Evolving Classification

In our approach, the *Evolving Classification* consists of two different modules: 1) *Creation of the Evolving Fuzzy Rules* and 2) *Web News Classification*. These modules are shown in Figure 1, where we can see that the fuzzy rules created by the first module are used by the classification module. Before explaining in detail

these modules, we will describe some characteristics of the *Evolving Fuzzy Rules* used in this approach.

The creation and update of the set of fuzzy rules is based on the *eClass0* classifier [55]. *eClass0* is a fuzzy rule-based (FRB) classifier which uses (fuzzy) rules which evolve from streaming data. In particular, *eClass0* possesses a zero-order Takagi-Sugeno consequent, so a fuzzy rule in our case has the following structure:

$$Rule_i = IF(A_1 \sim \ Prot_1)\ AND \ldots AND(A_n \sim \ Prot_n)$$
$$THEN\ \ Category\ =\ Category_j$$

where i represents the number of rule; n is the number of input variables (terms in the corpus); the A_i stores the *tf-idf* of the term i of the article (A) to classify, and the $Prot_i$ stores the *tf-idf* value of the term i of one of the prototypes (cluster center) of the corresponding class (*Category*). $Category_m \in \{$set of different categories$\}$.

An important aspect to consider is the possible interpretation by human of these (evolving) fuzzy rules. In addition, by analyzing these rules, we can get detect the most relevant terms of the different categories, and how these terms change over time.

In the next two subsections we will detail how these rules are created/updated (section 5.1), and used to classify a new article (section 5.2).

5.1. Creation of the Evolving Fuzzy Rules

This module is responsible for updating the number of rules and its attributes according to the text content of the news articles collected. In this approach, as soon as a web news article is collected and analyzed, its corresponding *tf-idf* values (per term) are sent to the *Creation of the Evolving Fuzzy Rules* module. The number of rules created by this module depends on the heterogeneity of the news articles in the same category.

As it is explained in [55], a prototype is a data sample that groups several samples which represent a certain class. The classifier is initialized with the first data sample, which is stored as a new rule. Based on the *potential* of the new data sample to become a prototype [9], it could form a new prototype or replace an existing one.

If we consider that the *new* news article is A_k and its category C_j, the fuzzy rules are updated (or created) by performing the following four steps: 1) Calculate the potential of A_k. 2) Update all the prototypes considering A_k. 3) Insert A_k as a

new prototype (of the Category C_m) if needed. 4) Remove existing prototypes if needed. These steps are explained in detail in the next subsections.

5.1.1. Calculate the potential of A_k

The potential (P) of the k^{th} news article (A_k) can be calculated by the equation 1, which represents a function of the accumulated distance between a news article and all the other k-1 articles in the data space [55]. The result of this function represents the *density* of the data that surrounds a certain article.

$$P(A_k) = \frac{1}{1 + \frac{\sum_{i=1}^{k} {}^{1} distance(A_k, A_i)}{k-1}} \tag{1}$$

where *distance* represents the distance between two news articles in the data space. Although different distance can be used to measure the similarity between two articles, we use cosine distance (equation 2) because it tolerates different samples to have different number of attributes (in this case, two articles could have a different number of terms since the corpus is constantly updated).

$$cosDist(A_k, A_p) = 1 - \frac{\sum_{j=1}^{n} A_{kj} x_{pj}}{\sqrt{\sum_{j=1}^{n} A_{kj}^2 \sum_{j=1}^{n} A_{pj}^2}} \tag{2}$$

where A_k and A_p represent the two news articles to measure its distance and n represents the number of different attributes in both samples.

However, if we use in this approach the equation 1 we need to store all the news articles, which contradicts to the requirement for real-time and on-line application needed in the proposed problem. For this reason, in [55] it is developed a recursive expression for the potential. This formula (equation 3) is as follows:

$$\begin{aligned} & P_k(A_k) = \frac{1}{2 - \frac{1}{(k-1)\sqrt{\sum_{j=1}^{n}(A_k^j)^2}} B_k}; k = 2, 3... \\ & where: \ B_k = \sum_{j=1}^{n} A_k^j b_k^j \ ; \ b_k^j = b_{(k-1)}^j + \sqrt{\frac{(z_k^j)^2}{\sum_{l=1}^{n}(A_k^l)^2}} \\ & and \ b_1^j = \sqrt{\frac{(A_1^j)^2}{\sum_{l=1}^{n}(A_1^l)^2}} \ ; \ j = [1, n+1]; \ P_1(A_1) = 1 \end{aligned} \tag{3}$$

where A_k represents the k^{th} news article and its corresponding label ($z = [A, Category]$). Using this expression, and this is one of the most important aspect

of this classifier, it is only necessary to calculate *(n+1)* values where *n* is the number of different terms of the corpus; this value is represented by b, where $b_k^j, j = [1, n]$ represents the accumulated value for the k^{th} articles.

5.1.2. Update all the prototypes considering A_k

The *density* of the data space surrounding certain prototype changes with the insertion of each new article, and the existing prototypes need to be updated. This update is fast since the recursive equation 3 is used.

5.1.3. Insert A_k as a new prototype (of the Category C_m) if needed

In this step, the potential of A_k is compared with the potential of all the existing prototypes. A new prototype is created if its value is higher than any other existing prototype, as shown in equation 4.

$$\exists i,\ i = [1, NumProt] :\ P(z_k) > P(Prot_i) \tag{4}$$

Thus, if A_k has high descriptive power and generalization potential, the classifier evolves by adding a new prototype, that is, a new rule in set of fuzzy rules.

5.1.4. Remove existing prototypes if needed

After adding a new prototype, we check whether any of the already existing prototypes are described *well* by the newly added prototype [55]. By *well* we mean that the value of the membership function that describes the closeness to the prototype is a Gaussian bell function chosen due to its generalization capabilities:

$$\exists i,\ i = [1, NumPrototypes] :\ \mu_i(A_k) > e^{-1} \tag{5}$$

The membership function between a data sample and a prototype is calculated as follows:

$$\mu_i(A_k) = e^{-\frac{1}{2}[\frac{cosDist(z_k, Prot_i)}{\sigma_i}]}\ ,\ i = [1, NumProt] \tag{6}$$

where $cosDist(A_k, Prot_i)$ represents the cosine distance between A_k and the i^{th} prototype; σ_i represents the spread of the membership function, which also symbolizes the radius of the zone of influence of the prototype. This spread is determined based on the scatter of the prototype. In order to calculate the scatter without storing all the news articles, this value can be updated (as shown in [55]) recursively by:

$$\sigma_i(k) = \sqrt{[\sigma_i(k-1)]^2 + \frac{[cosDist^2(Prot_i, z_k) - [\sigma_i(k-1)]^2]}{k}} \quad (7)$$

where *k* is the number of news articles considered; *cosDist(*P_i, A_k*)* is the cosine distance between the i^{th} prototype and the new news article.

5.2. Web News Classification

This module classifies a new news article (A_z) into one of the categories previously analyzed. Since a category is represented by one or more prototypes, A_z is compared to all the prototypes by using cosine distance and the smallest distance determines the closest similarity. This aspect is considered in equation 8.

$$\begin{aligned} &Class(A_z) = Class(Prot^*); \\ &Prot^* = MIN_{i=1}^{NumProt}(cosDist(Prot_i, A_z)) \end{aligned} \quad (8)$$

where A_z represent the (non-categorized) web news article to classify, $NumProt$ determines the number of existing prototypes (number of rules), $Prot_i$ represents the i_{th} prototype, and $cosDist$ represents the cosine distance between two news articles.

6. Experimental Design and Results

In order to evaluate the presented approach, we have collected hundreds of news articles in English language from the New York Times online newspaper. This collection has been done by using the *NYT API* as we have explained in section 4. It should be emphasized that this approach has been totally designed to be applied in real-time. However, we have created several data sets of hundreds of web news articles in order to know the accuracy of our approach and to have comparable results with other off-line techniques.

6.1. Data Sets

For the purpose of this research, we have collected hundreds of web *NYT* news articles which are already categorized in seven different topics: *Art, Business, Health, Science, Sports, Technology* and *Travel*. In total, we have collected 3500 categorized web news articles: 500 news articles for each of the 7 categories. We want to emphasize again that using our approach the number of categories does not need to be predefined and it can changes over time.

In addition, to evaluate the proposed approach using different numbers of categories, the following 5 different sets of data which combine two or more categories have been created: 1) *Health vs. Science*, 2) *Science vs. Techonology*, 3) *Health vs. Science vs. Sports*, 4) *Business vs. Health vs. Science vs. Sports*, and 5) *Arts vs. Business vs. Health vs. Science vs. Sports vs. Travel*.

The main terms of these 3500 articles (taking into account the different categories) were extracted as it was proposed in the *Term Extraction* module. However, since the number of different terms is very high (more than 6000 different terms), we have applied an addition terms reduction technique. This reduction has been done taking into account that the *tf-idf* of several terms is very low. For this reason, we have removed those terms with a low *tf-idf* value in all the documents. Specifically, we sum the *tf-idf* values of a particular term in the collection of documents. If this value is lower than a threshold (*pruning threshold*), it is removed from the data set. In this research, we have used several *pruning threshold* (from 0.3 to 2.5) in order to evaluate its influence in the final results. As higher the threshold is, as smaller the final data set is. We should consider that the number of terms removed using a *pruning threshold* value higher than 1 is very high.

To get an idea about how the size of the corpus decreases based on the threshold that is applied to reduce the original corpus, Table 1 summarizes the number of terms of the 5 sets of categories using different *pruning thresholds*. For example, we can observe that if there is no reduction (original data set), the number of different terms using 6 categories is 6112; however, if we prune the number of terms using a threshold of 2.5, this number is drastically reduced to 38 terms. In addition, after applying this terms reduction, we remove a very small number of articles which were represented only by terms with value 0.

6.2. Results

In order to measure the performance of the proposed approach in the previous 5 data sets (reduced by different *pruning thresholds* values), the well-established technique of *10-fold cross-validation* is chosen. The different pruned data sets (training set) are divided into 10 disjoint subsets with equal size. Each of the 10 subsets is left out in turn for evaluation. Our approach does not need to work this kind of validation; however, it has been evaluated in this mode to get an idea about its performance.

The results are shown in Figure 2, where the different lines (different colors) represent different combination of categories. Each combination has been evaluated by using 11 different pruning thresholds (the x-axis indicates this value).

Table 1: Number of terms (Corpus) of the different data sets taking into account the data reduction using *tf-idf* (*Pruning Threshold*).

	Number of Terms						
		Pruning Thresholds					
Data Sets	Original	<0.3	<0.5	<1	<1.5	<2	<2.5
Health-Science	2765	1843	857	279	111	59	34
Science-Technology	2892	1857	823	273	109	61	31
Health-Science-Sports	3759	1883	863	289	112	58	35
Business-Health--Science-Sports	4280	1896	842	273	112	55	36
Arts-Business-Health--Science-Sports-Travel	6112	1902	841	276	118	62	38

Figure 2 shows that the percentage of news articles correctly classified using the original data (not pruned) can be improved by removing those terms which are not representative. For example, if we classify a news article into only two categories (Science and Technology) using a *pruning threshold* value of 2, the number of terms decrease drastically from 2892 to 61. However, the percentage of news correctly classified improves from 53% to 73%. Thus, this pruning process is essential in this environment.

In general, we can conclude that using this approach, the results are better when the terms reduction is high (using a *pruning threshold* higher than 1.5). The main reason for this behavior is that when we use too many words for categorizing a news article, those words really important are "hidden" by the total set of words. This aspect is important since we do not need all the terms of the news articles to be able to classify them with a high accuracy. Thus, the choice of the *pruning threshold* has a crucial impact in the results obtained. If we use this research with other categories, a briefly study about the impact of the *pruning threshold* in the classification could done as part of the approach.

As we supposed, if the number of categories (classes) is more than two, the percentage of news correctly classified decreases. For example, using 6 different categories and a pruning thresholds of 2.5, we obtain a classification accuracy of almost 40%. To analyze this results we should take into account the reduced

number of terms (38) analyzed for this task.

6.3. Comparison with other classifiers

For evaluating the performance of our approach, we compare the proposed classifier (*eClass0*) with the following well-known classifiers:

1. *C4.5* classifier [56]: a well-known decision tree induction algorithm (its predecessor is ID3 algorithm).
2. *Naive Bayes* classifier (*NB*) using estimator classes [57]: simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions.
3. *Artificial Neural Network* (*ANN*): A multi-layer perceptron trained with the back-propagation algorithm, in which the nodes are all sigmoid. Also, the number of neurons is modified according to the number of attributes and classes (in particular: (attributes + classes) / 2).
4. *Support Vector Machines* (*SVM*) [58]: kernel based classifier which implements John Platts sequential minimal optimization algorithm.
5. *K Nearest Neighbor Classifier* (*k-NN*) [59]: an instance-based learning technique based on closest training examples in the feature space.

In addition, we have chosen an incremental classifier (*NB Incremental*) in order to evaluate *eClass0* with other incremental classifier which can be used in real time. However, it is important to remark that *eClass0* is recursive, the rule-base are constantly updated, and the obtained rules are subject to interpretation.

This comparison is done for the 5 different combination of categories and using a pruning threshold value of 1,85 (with which, in general, a better accuracy is obtained). In addition, it is important to remark that in this section, we are comparing only the classifiers since the term extraction module is the same for all the classifiers.

In table 2 we observe that the percentage of news articles correctly classified by our proposed classifier (*eClass0*) are similar to those obtained by *NB Incremental* and C4.5. Nevertheless, the difference between *eClass0* and ANN, K-NN and SVM is considerable. However, the classifier based on ANN needs a great deal of neurons and the training process is very time consuming, computationally expensive and impractical in implementation in the environment proposed in this work. Since we are looking for a classifier which can process streaming data in real-time, only the incremental classifiers satisfy this requirement. In addition, as it has already been explained, *eClass0* is a good working alternative for several

Table 2: Classification rates: eClass0 vs other well-known classifiers. Abbreviations of the categories: H (Health), Te (Technology), Sc (Science), Sp (Sports), B (Bussines), A (Arts), Tr (Travel).

		Classifiers and classification rate (%)						
DataSet (Combinat. Categories)	Pruning Threshold	Incremental Classif.		Non-Incremental Classifiers				
		eClass0 (Approach)	NB (Incr.)	C4.5	NB	ANN	K-NN (k=1)	SVM
H-Sc	1.85	72.2	68.3	69.8	70.3	83.3	73.4	74.9
Sc-Te	1.85	78.3	81.7	77.8	87.4	88.3	86.4	90.8
H-Sc-Sp	1.85	64.4	65.4	65.8	66.4	79.3	78.2	76.6
B-He-Sc-Sp	1.85	52.3	53.0	59.0	64.3	63.8	70.4	66.6
A-B-H- -Sc-Sp-Tr	1.85	35.7	36.3	40.1	38.5	45.9	50.4	48.2

reasons: 1) can cope with huge amounts of data because it does not need to store the entire data streams in the memory, 2) is open and the rule-base evolves, 3) is computationally simple and efficient as it is recursive and one pass.

6.4. Evolving Fuzzy Rules in Web News Classification

As we have already detailed, a specific category of this environment can be represented by one or several rules, depending on the heterogeneity of the news articles that represent the same category. Thus, a class could be represented by one or several prototypes. The different rules (prototypes) that represent a category are constantly updated.

In order to clarify this idea (which is the main contribution of this paper), let us consider that news article from two categories (*Health* and *Technology*) are collected. In this case, the first news article collected creates the first prototype. Let us suppose that this news article is categorized as *Health*, the first rule could be something like this:

$$Evolving Fuzzy Rule - Example1:$$

$$A)\ Rule Health1:$$
$$IF\ (\text{“}home''\ \sim\ 0,397)\ AND\ (\text{“}drug''\ \sim\ 0,2389)\ AND \ldots$$
$$\ldots AND\ (\text{“}doctor''\ \sim\ 0,1876)\ THEN\ Category =' HEALTH'$$

However, this set of rules evolves by collecting and processing new news articles. In such a way, the number of rules and terms constantly changes. Taking into account the previous example, let us consider the following example (*Example2*) in which the category named as 'Health' is represented by different rules (2 different prototypes) and the category *Technology* is represented by only one rule. We can observe the high interpretability of the rules which makes that different categorizes can be easily analyzed.

$EvolvingFuzzyRules - Example2:$

$A)\ RuleHealth1:$
$IF\ (\text{“}home\text{“}\ \sim\ 0,2886)\ AND\ (\text{“}drug\text{“}\ \sim\ 0,3468)\ AND \ldots$
$\ldots AND\ (\text{“}disease''\ \sim\ 0,1876)\ THEN\ Category =' HEALTH'$

$RuleHealth2:$
$IF\ (\text{“}drug\text{“}\ \sim\ 0,3260)\ AND\ (\text{“}doctor\text{“}\ \sim\ 0,4503)\ AND \ldots$
$\ldots AND\ (\text{“}patient''\ is\ 0,3973)\ THEN\ Category =' HEALTH'$

$B)\ RuleTechnology:$
$IF\ (\text{“}video\text{“}\ \sim\ 0,1638)\ AND\ (\text{“}Googl\text{“}\ \sim\ 0,4022)\ AND \ldots$
$\ldots AND\ (\text{“}compani''\ \sim\ 0,3250)\ THEN\ Category =' TECHNOLOGY'$

Although this is an example, the real number of rules that are created from the previous data sets (and *pruning threshold* = 1.85) is shown in Table 3. This Table shows how the total number of Evolving Fuzzy Rules that are created using our approach is very small. In general, it is remarkable the small number of rules which are created. This aspect is important since the data to be stored by our approach is much reduced.

7. Conclusions and Future work

The research presented in this paper is mainly related with two fields: Big data and Web news mining. We propose an approach in the field of text mining (in particular, web news mining), but it is always considered that the amount of news to process is huge.

In this sense, our proposal is an evolving approach for classifying different web news articles into various topic areas based on the text content of the articles. Since news websites are daily overwhelmed with plenty of news articles,

Table 3: Number of Fuzzy Rules which are created by our approach (H-Health, Sc-Science, Sp.-Sports, B.-Business, A.-Arts, Tr.-Travel,Te.-Technology)

	Categories							
Data Sets	H.	Sc.	Sp.	B.	A.	Tr.	Te.	Total
Health-Science	2	2						4 Rules
Science-Technology		4					5	9 Rules
Health-Science-Sports	7	6	3					16 Rules
Business-Health--Science-Sports	6	3	4	5				18 Rules
Arts-Business-Health--Science-Sports-Travel	7	8	5	6	3	3		32 Rules

one of the main advantages of the proposed approach is that it can cope with huge amounts of news in real-time. In addition, since the web news articles change everyday, we have proposed an evolving classifier which also changes constantly. This classifier is based on Evolving Fuzzy Systems (EFS) and the model that describes a specific topic area changes according to the change in the text content of their articles. In addition, our proposal not only classifies news articles, but it also creates human interpretable models of the different categories.

In order to get the relevance of the different terms of the web news, we have seen that *tf-idf* is a simple but powerful and efficient numeric measure for this task. Related to the proposed (evolving) classifier, it is important to highlight that it is very simple and it works very fast. Also, the proposed classifier is one pass, non-iterative, recursive and it can be used in an interactive mode. Thus, this method can cope with huge amounts of news and process them quickly. Although the amount of terms from the articles is huge, we can extract the most important terms with no need to store all the news in memory. The approach has been successfully tested using real on-line news, and according to the results, we can conclude that it performs as well as other well-known incremental classifier, specially using a (very) small number of terms. In this sense, the value of the *pruning threshold* is essential in order to achieve good results.

Finally, it is important to highlight that although our proposal is focused on the web news mining, this approach can be used in other different areas in which the process of huge amounts of data in real time is needed. For example, since a

web news is represented by a set of terms, our approach can be used to categorize social networks messages (ie, tweets) in real-time.

Acknowledgments. This work has been supported by the Spanish Government under i-Support (Intelligent Agent Based Driver Decision Support) Project (TRA2011-29454-C03-03).

References

[1] Big data: The next frontier for innovation, competition, and productivity, Tech. Rep., 2011.

[2] CINIC, Statistical Report on Internet Development in China, Tech. Rep., China Internet Network Information Center, 2014.

[3] R. Kosala, H. Blockeel, Web Mining Research: A Survey, SIGKDD Explor. Newsl. 2 (1) (2000) 1–15, ISSN 1931-0145.

[4] R. Barzilay, K. R. McKeown, Sentence Fusion for Multidocument News Summarization, Comput. Linguist. 31 (3) (2005) 297–328.

[5] P. Sutheebanjard, W. Premchaiswadi, Disambiguation of Thai personal name from online news articles, in: Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, vol. 3, 302–306, 2010.

[6] L.-F. Hsu, Mining on Terms Extraction from Web News, in: J.-S. Pan, S.-M. Chen, N. Nguyen (Eds.), Computational Collective Intelligence. Technologies and Applications, vol. 6421 of *LNCS*, Springer, ISBN 978-3-642-16692-1, 188–194, 2010.

[7] M. Kabadjov, J. Steinberger, R. Steinberger, Multilingual Statistical News Summarization, in: T. Poibeau, H. Saggion, J. Piskorski, R. Yangarber (Eds.), Multi-source, Multilingual Information Extraction and Summarization, Theory and Applications of Natural Language Processing, Springer Berlin Heidelberg, ISBN 978-3-642-28568-4, 229–252, 2013.

[8] D. Kim, D. Kim, S. Kim, M. Jo, E. Hwang, SNS-based Issue Detection and Related News Summarization Scheme, in: Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, ICUIMC '14, ISBN 978-1-4503-2644-5, 114:1–114:7, 2014.

[9] P. Angelov, D. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34 (1) (2004) 484 – 498, ISSN 1083-4419.

[10] P. Angelov, D. Filev, N. Kasabov, Evolving intelligent systems: methodology and applications, IEEE Press Series on Computational Intelligence, Wiley-Blackwell, 2010.

[11] J. A. Iglesias, P. Angelov, A. Ledezma, A. Sanchis, Human Activity Recognition Based on Evolving Fuzzy Systems, International Journal of Neural Systems 20 (5) (2010) 355–364.

[12] P. Sadeghi-Tehran, P. P. Angelov, R. Ramezani, A Fast Recursive Approach to Autonomous Detection, Identification and Tracking of Multiple Objects in Video Streams under Uncertainties., in: IPMU (2), vol. 81 of *Communications in Computer and Information Science*, Springer, 30–43, 2010.

[13] J. A. Iglesias, P. Angelov, A. Ledezma, A. Sanchis, Creating Evolving User Behavior Profiles Automatically, Knowledge and Data Engineering, IEEE Transactions on 24 (5) (2012) 854–867, ISSN 1041-4347.

[14] P. P. Angelov, I. Skrjanc, S. Blazic, Robust evolving cloud-based controller for a hydraulic plant, in: EAIS, 1–8, 2013.

[15] J. A. Iglesias, I. Skrjanc, Applications, results and future direction (EAIS 12), Evolving Systems 5 (1) (2014) 1–2.

[16] D. Kangin, P. Angelov, J. Iglesias, A. Sanchis, Evolving Classifier TEDA-Class for Big Data, in: INNS Conference on Big Data 2015, 2015.

[17] W. Mason, J. Vaughan, H. Wallach, Computational social science and social computing, Machine Learning 95 (3) (2014) 257–260, ISSN 0885-6125.

[18] K. J. Edwards, M. M. Gaber, Astronomy and Big Data: A Data Clustering Approach to Identifying Uncertain Galaxy Morphology, Springer Publishing Company, Incorporated, ISBN 331906598X, 9783319065984, 2014.

[19] N. Savage, Bioinformatics: Big data versus the big C, Nature 509 (7502) (2014) S66–S67.

[20] K. Tang, M. Lin, F. L. Minku, X. Yao, Selective negative correlation learning approach to incremental learning., 2796–2805, 2009.

[21] W.-F. Hsiao, T.-M. Chang, An incremental cluster-based approach to spam filtering, Expert Systems with Applications 34 (3) (2008) 1599 – 1608, ISSN 0957-4174.

[22] C. P. Chen, C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, Information Sciences 275 (0) (2014) 314 – 347, ISSN 0020-0255.

[23] R. Feldman, I. Dagan, Kdt - knowledge discovery in texts, in: International Conference on Knowledge Discovery (KDD), 112–117, 1995.

[24] S. M. Weiss, N. Indurkhya, T. Zhang, F. Damerau, Text mining: predictive methods for analyzing unstructured information, Springer, 2010.

[25] S.-H. Liao, P.-H. Chu, P.-Y. Hsiao, Data mining techniques and applications A decade review from 2000 to 2011, Expert Systems with Applications 39 (12) (2012) 11303–11311.

[26] A. Tan, Text Mining: The state of the art and the challenges, in: In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases, 65–70, 1999.

[27] A. T. Sadiq, S. M. Abdullah, Hybrid Intelligent Techniques for Text Categorization, International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol 2 (2014) 23–40.

[28] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, D. C. L. Ngo, Text mining for market prediction: A systematic review, Expert Systems with Applications 41 (16) (2014) 7653 – 7670.

[29] G. Oberreuter, J. D. Velsquez, Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style, Expert Systems with Applications 40 (9) (2013) 3756 – 3763.

[30] Y. Lu, P. Zhang, J. Liu, J. Li, S. Deng, Health-related hot topic detection in online communities using text clustering, PloS one 8 (2) (2013) e56221.

[31] C. Aggarwal, C. Zhai, A Survey of Text Clustering Algorithms, in: Mining Text Data, Springer US, 77–128, 2012.

[32] A. Sunikka, J. Bragge, Applying text-mining to personalization and customization research literature Who, what and where?, Expert Systems with Applications 39 (11) (2012) 10049 – 10058.

[33] U. Y. Nahm, R. J. Mooney, A Mutually Beneficial Integration of Data Mining and Information Extraction., in: AAAI/IAAI, AAAI Press / The MIT Press, 627–632, 2000.

[34] B. Lent, R. Agrawal, R. Srikant, Discovering Trends in Text Databases., in: Proceedings of KDD, 227–230, 1997.

[35] D. Godfrey, C. Johns, C. D. Meyer, S. Race, C. Sadek, A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets, CoRR abs/1408.5427, URL http://arxiv.org/abs/1408.5427.

[36] S. L. Lo, D. Cornforth, R. Chiong, Identifying the high-value social audience from Twitter through text-mining methods, in: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1, Springer, 325–339, 2015.

[37] R. Nikhil, N. Tikoo, S. Kurle, H. S. Pisupati, G. Prasad, A Survey on Text Mining and Sentiment Analysis for Unstructured Web Data, in: Journal of Emerging Technologies and Innovative Research, vol. 2, JETIR, 2015.

[38] M. Mathioudakis, N. Koudas, TwitterMonitor: Trend Detection over the Twitter Stream, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, ACM, New York, NY, USA, ISBN 978-1-4503-0032-2, 1155–1158, doi:10.1145/1807167.1807306, URL http://doi.acm.org/10.1145/1807167.1807306, 2010.

[39] D. Kim, D. Kim, E. Hwang, S. Rho, Twittertrends: a spatio-temporal trend detection and related keywords recommendation scheme, Multimedia Systems 21 (1) (2014) 73–86.

[40] H. Sayyadi, S. Salehi, H. AbolHassani, Survey on News Mining Tasks, in: Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, Springer Netherlands, 219–224, 2007.

[41] J. Li, J. Li, J. Tang, A flexible topic-driven framework for news exploration, in: Proceedings of KDD, 2009.

[42] D. E. Appelt, Introduction to Information Extraction, AI Commun. 12 (3) (1999) 161–172.

[43] J. Piskorski, M. Atkinson, J. Belyaeva, V. Zavarella, S. Huttunen, R. Yangarber, Real-time Text Mining in Multilingual News for the Creation of a Pre-frontier Intelligence Picture, in: ACM SIGKDD Workshop on Intelligence and Security Informatics, ACM, 1–9, 2010.

[44] J. Yoon, Detecting weak signals for long-term business opportunities using text mining of Web news, Expert Systems with Applications 39 (16) (2012) 12543 – 12550.

[45] S. Malhotra, A. Dixit, An Effective Approach for News Article Summarization, International Journal of Computer Applications 76 (16) (2013) 5–10.

[46] S. G. Chowdhury, S. Routh, S. Chakrabarti, News Analytics and Sentiment Analysis to Predict Stock Price Trends, International Journal of Computer Science and Information Technologies 5 (3) (2014) 3595–3604.

[47] J. Liu, P. Dolan, E. R. Pedersen, Personalized News Recommendation Based on Click Behavior, in: Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI '10, ISBN 978-1-60558-515-4, 31–40, 2010.

[48] G. De Francisci Morales, A. Gionis, C. Lucchese, From Chatter to Headlines: Harnessing the Real-time Web for Personalized News Recommendation, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, ACM, New York, NY, USA, ISBN 978-1-4503-0747-5, 153–162, 2012.

[49] I. Mani, Automatic Summarization, John Benjamins Publishing Co., 2001.

[50] D. K. Evans, J. L. Klavans, K. R. McKeown, Columbia Newsblaster: Multilingual News Summarization on the Web, in: Demonstration Papers at HLT-NAACL 2004, Association for Computational Linguistics, 1–4, 2004.

[51] M. Litvak, M. Last, M. Friedman, A New Approach to Improving Multilingual Summarization Using a Genetic Algorithm., in: ACL, The Association for Computer Linguistics, 927–936, 2010.

[52] I. Mierswa, M. Scholz, R. Klinkenberg, M. Wurst, T. Euler, YALE: Rapid Prototyping for Complex Data Mining Tasks, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 935–940, 2006.

[53] M. F. Porter, Snowball: A language for stemming algorithms, URL http://snowball.tartarus.org/texts/, 2001.

[54] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management 24 (5) (1988) 513–523.

[55] P. P. Angelov, X. Zhou, Evolving Fuzzy-Rule-Based Classifiers From Data Streams, IEEE T. Fuzzy Systems 16 (6) (2008) 1462–1475.

[56] J. R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc., 1993.

[57] G. H. John, P. Langley, Estimating Continuous Distributions in Bayesian Classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-385-9, 338–345, URL http://dl.acm.org/citation.cfm?id=2074158.2074196, 1995.

[58] J. C. Platt, Advances in Kernel Methods, chap. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, MIT Press, Cambridge, MA, USA, ISBN 0-262-19416-3, 185–208, URL http://dl.acm.org/citation.cfm?id=299094.299105, 1999.

[59] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.