An FAR-SW based Approach for Webpage Information Extraction

Zhan Bu[†‡], Chengcui Zhang[‡], Zhengyou Xia[†], Jiandong Wang[†]

[†]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China*

E-Mail: {buzhan, zhengyou_xia, aics} @nuaa.edu.cn

[‡]*Computer and Information Sciences, The University of Alabama at Birmingham, USA*

E-Mail: {zhanb, zhang}@cis.uab.edu

Abstract Automatically identifying and extracting the target information of a webpage, especially main text, is a critical task in many web content analysis applications, such as information retrieval and automated screen reading. However, compared with typical plain texts, the structures of information on the web are extremely complex and have no single fixed template or layout. On the other hand, the amount of presentation elements on web pages, such as dynamic navigational menus, flashing logos, and a multitude of ad blocks, has increased rapidly in the past decade. In this paper, we have proposed a statistics-based approach that integrates the concept of fuzzy association rules (FAR) with that of sliding window (SW) to efficiently extract the main text content from web pages. Our approach involves two separate stages. In Stage 1, the original HTML source is pre-processed and features are extracted for every line of text; then, a supervised learning is performed to detect fuzzy association rules in training web pages. In Stage 2, necessary HTML source preprocessing and text line feature extraction are conducted the same way as that of Stage 1, after which each text line is tested whether it belongs to the main text by extracted fuzzy association rules. Next, a sliding window is applied to segment the web page into several potential topical blocks. Finally, a simple selection algorithm is utilized to select those important blocks that are then united as the detected topical region (main texts). Experimental results on real world data show that the efficiency and accuracy of our approach are better than existing Document Object Model (DOM)-based and Vision-based approaches.

Keywords *Information extraction; Statistics-based; Fuzzy association rule; Sliding window; Topical region*

1 Introduction

With the advance of information technology as well as the booming of the Internet, people can easily acquire any information they want. However, compared with typical plain texts, the structures of information on the web are quite different and much more diversified, forming a large, distributed, and heterogeneous database environment. These data are extremely complex, with no single fixed template or layout, and are usually called semi-structured data. On the other hand, the evolution of web browser technology in the past decade has caused a significant growth in the number of presentation elements on web pages. Web content in most popular sites exploits these browser features, such as dynamic navigational menus, flashing logos, a multitude of ad blocks, rich headers, and footers. In the work of Gibson et al. (2005), they estimate that layout presentation

elements constitute 40% to 50% of all Internet content and this volume has been increasing approximately 6% yearly.

In such cases, identifying and extracting the target information of a webpage, especially main texts, is a critical task in many web content analysis applications. To name just a few, information retrieval systems, knowledge base systems, and search engines can benefit from a cleaner store of the webpage's data which in turn can provide more accurate search results. Screen reading software (such as JAWS, Window-Eyes, and Dolphin Supernova) may focus on the main content and skip templates and other irrelevant content (Theofanos and Redish 2003). Some small screen devices, such as modern mobile phones, can increase readability by focusing on displaying the main body text only.

However, due to the abovementioned reasons, automatically identifying and extracting information from semi-structured web sources is an important research topic. In recent years, a large number of researches have addressed this problem, and many important research results have been put forward (Koch 2001; Kao et al. 2005; Cai 2003; Kang et al. 2010; Alexjc 2007; Zhou et al. 2009). Differentiated by their scopes, these works can be categorized into Document Object Model (DOM) based, vision-based, and statistics-based approaches:

a) DOM-based segmentation approaches need to first construct the DOM tree structure from the HTML source of the web page, and then, extract the important content by pruning the DOM branches. However, such DOM tree processing tasks are very time-consuming; therefore, they cannot satisfy those online analysis applications with a real-time requirement.

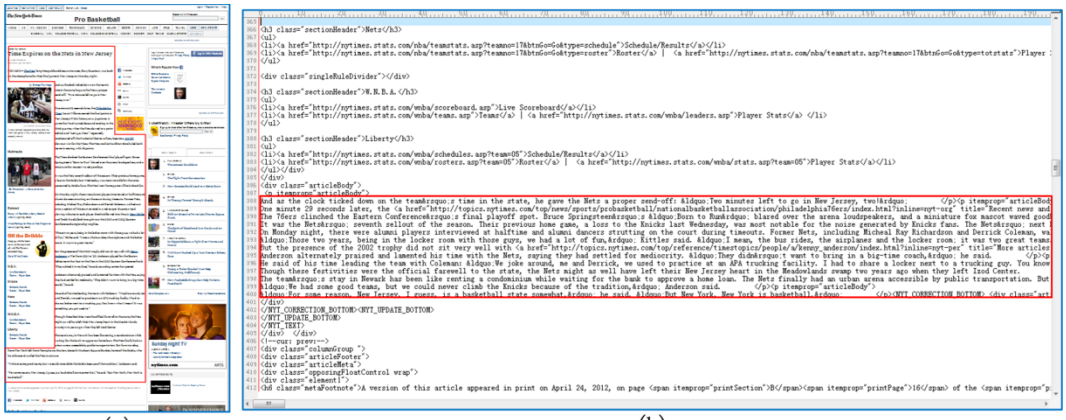
b) Vision-based page segmentation (VIPS) makes full use of page layout features such as font, color and size. It firstly extracts all the suitable nodes from the DOM tree, and then finds the separators, which denote the horizontal or vertical lines in a web page that visually do not cross any node. However, this kind of algorithm needs a large amount of computations to analyze the web page, and has inherent dependence on the data sources. For example, the regular expressions used to extract line separators from HTML codes are code-source specific, and there is no single universal line extractor.

c) Statistics-based segmentation approaches aim to extract text paragraphs from large chunks of HTML code, without knowing its structure or the tags used. It uses statistics or machine learning methods to save time and effort.

Among the three methods, the first two both need to render the page, and exploit specific extraction strategies during the extraction process. For example, for DOM based method, one needs to decide which tags are used to build the DOM tree and which leaf nodes are potential story blocks. Though these approaches usually provide reasonably good results, they come at a high computational cost. While in statistics-based approaches, the model of web pages can be acquired without rendering the page or any human intervention; therefore, it has a wider practical utilization prospect in the information extraction area.

In this paper, we design a statistics-based approach with that integrates the concepts of fuzzy association rule (FAR) with sliding window (SW) to efficiently and effectively extract the main text content from web pages. Our method is based on a practical observation that the main text of a web page usually occupies the center of the web page with multiple adjacent, relatively long text paragraphs, especially for web pages of news, blogs, articles, etc. What is more, the main text in a news page usually contains lots of texts with few links or images. Therefore, the content types and compositions in a paragraph are useful indicators of importance of that paragraph. Fig. 1 gives a news web page from New York Times News, and its original HTML source. In Fig. 1(a), the news content is highlighted by a red-lined polygon. As shown in Fig. 1(b), the extracted main text (outlined in a red box), which only constitutes 17.2% of the HTML source, is located around the middle part of the HTML source. In addition, the main text part of the code has a high pure-text density, with very few links or images. On the basis of these observations, we propose a new approach that has a high throughput in processing web page documents, while still producing satisfactory results. First, our method analyzes every text line in the original HTML source and use Fuzzy Association Rules to determine whether it belongs to the main text. Then, a Sliding Window technique is applied to "smooth out" adjacent text lines and to extract potential main text blocks. Finally, topical regions can be formed by applying a simple selection algorithm.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the FAR-SW based approach for extracting main texts from web pages. In order to evaluate our approach, we conduct a few experiments on real world data sets, and the comparison results are presented in Section 4. Finally, we conclude the paper in Section 5.



(a) (b)

Fig. 1 An example of a News page in www.nytimes.com and its HTML source

2 Related works

There have been a number of studies that analyze an HTML page in order to extract the target information (e.g., main texts) from the pages. It is common to differentiate by their scope, which creates the notions of page-level and site-level approaches. The former gains its generality at the cost of slightly worse results, but requires little maintenance, has low setup costs and broad usability since the approach works independently of the site's design. The latter requires some non-trivial amount of example data to build a model or rules that are specific for the pages from that site. As it is tailored for a specific group, the results are generally better, but come at the cost of high maintenance, high setup costs and limited usability due to the wrappers that are created to exploit particularities of each site's design.

As web pages can be parsed as a tree structure called Document Object Model (Koch 2001) by a standard HTML parser, most existing approaches are based on the DOM tree structure constructed from the HTML source of the web page. Important content can be extracted by analyzing the organization structure of the DOM tree and the properties of the DOM nodes. Gupta et al. (2003) proposed a DOM-based model to extract main texts from HTML web pages. Their approach, working with the Document Object Model tree as opposed to raw HTML markup, can be used to perform main text extraction, identifying and preserving the original data instead of summarizing it. Kao et al. proposed WISDOM (Kao et al. 2005) system, which applies Information Theory to DOM tree in order to build the intra-page informative structure of web pages.

DOM was initially designed to render the web page rather than to describe the semantic structure of its content, therefore, it is not an optimal option for our task because constructing DOM trees is a time-consuming task, as shown by (Koch 2001; Kao et al. 2005), and the accuracy is not guaranteed when determining which DOM branches include the main content. Many heuristic methods were proposed in order to address the above problems. For example, a vision-based page segmentation algorithm (VIPS), proposed by Cai et al. (2003), segments a web page into semantically related content blocks based on its visual presentation. Compared with traditional DOM based segmentation methods, VIPS makes full use of page layout features to reconstruct the structure of a page. Each node in the extracted content structure corresponds to a block of coherent content in the original page. Kang et al. developed a new method, named REPB (Kang et al. 2010), to detect key patterns in a Web page and generates virtual nodes to correctly segment nested blocks. REPB generates a sequence from the simplified DOM tree by using the tags of the child nodes of the root node; then, it finds the key patterns from the sequence and recognizes candidate blocks by matching the sequence with the key patterns; finally, REPB generates blocks in a page by modifying the DOM tree into a more deeply hierarchical structure by introducing virtual nodes. Though vision-based heuristic methods may increase the accuracy of the original DOM based approaches, they are applied on the site level, which means that they require visual data from browser rendering engines.

The above methods all need parse the HTML source into a DOM tree. In addition to that, computing spatial and visual properties is also time consuming, making those approaches fail to satisfy those online applications with a real-time requirement. Recently, some statistics-based approaches have been proposed. The work in (Alexjc 2007) uses the text density of each paragraph (the ratio of the length of plain text to that of HTML source required to describe it). Then a neural

network is used to decide if the line is part of the content (main text). However, it requires additional work to manually label a large number of paragraphs from various web pages in training the neural network model. Moreover, its performance highly depends on the comprehensiveness of the labeled paragraphs. Zhou et al. process the HTML source as a paragraphed text string directly and extract the main text content by only analyzing the word count of text paragraphs (Zhou 2009). Suppose $P' = \{p_k, p_{k+1}, ..., p_{N'}\}$ represents the set of main text paragraphs in HTML source, and n''_{long} and n''_{short} are two global thresholds. P' must satisfy the following two conditions: 1) the length of every p_i must be greater than n''_{short}; 2) there exists at least one paragraph whose length is greater or equal to n''_{long}. However, using global thresholds (n''_{long} and n''_{short}) to filter texts is problematic at best. Both false positives and false negatives can be introduced during hard thresholding. To address this problem, we propose a new FAR-SW based approach, which can achieve a reasonably good accuracy for identifying the main text of web pages and is much faster compared with existing approaches.

3 FAR-SW based approach

The main tasks of our approach are 1) to identify potential topical lines in the original HTML source; 2) to segment a web page into several potential topical blocks. Fig. 2 shows an overview of our approach, which involves two separate stages: fuzzy association rule detection from training pages and main text extraction by matching the fuzzy association rules on testing pages. In Stage 1, the original HTML source needs necessary preprocessing and features are extracted for each text line; then, a supervised learning is performed to detect fuzzy association rules in training pages. In Stage 2, similar preprocessing and text feature extraction are performed, after which each text line is tested whether it belongs to the main text, i.e., how likely it is a topical line, by the extracted fuzzy association rules. Next, a sliding window is applied to topical lines and segments the web page into several potential topical blocks. Finally, a simple selection algorithm is utilized to select those important blocks that are then united as the detected topical regions (main text).

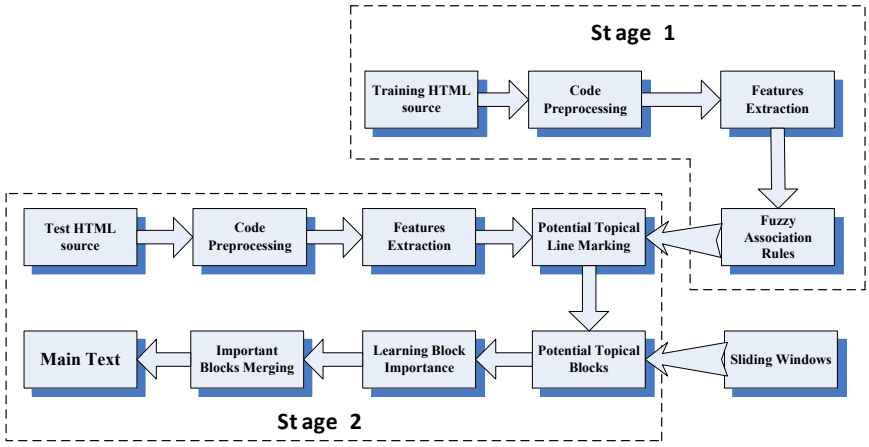


Fig. 2 Overview of our approach

Some notations used in our approach are described as follows:

D : The original database;

n : The number of topical lines in D ;

N : The number of text lines in D ;

T_i : The i^{th} topical line in D ($1 \le i \le n$);

p_i : The i^{th} text line in D ($1 \le i \le N$);

m : The number of items in D ;

I_j : The j^{th} item (feature) ($1 \le j \le m$);

h_j : The number of fuzzy regions for I_j ;

R_{jl} : The l^{th} fuzzy region of I_j $(1 \le l \le h_j)$;

v_{ij} : The value of I_j in T_i;

f_{ijl} : The membership value of v_{ij} in region R_{jl};

3.1 Code preprocessing

In today's information storage and retrieval applications, the growth in presentation elements increases difficulty in extracting relevant content. For example, tokens, phrases, and named-entities in advertising sections and footnotes become increasingly indistinguishable from those in the title and the body sections of a news article. To improve the accuracy of extraction results, those presentation elements need to be removed, as shown below:

1) Get the text between the pair of <BODY> tags;
2) Delete all blank lines and redundant white-spaces;
3) Delete HTML tags listed in Table 1, because the contents between which are always noise information.

After the above three steps, we obtain the web page HTML source with very little noise information.

Table 1 Some useless tags in HTML source files

Useless HTML tags
<a>, <script>, <noscript>, <style>, <meta>, <!-->, <param>, <button>, <select>, <optgroup>, <option>, <label>, <textarea>, <fieldset>, <legend, <input>, <image>, <map>, <area>, <form>, <iframe>, <embed>, <object>

3.2 Feature extraction

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets, within the web page content. HTML tags most commonly come in pairs: the first tag in a pair is the start tag, and the second tag is the end tag. Frequently, some attributes are also included in the start tag. In between these tags web designers can add texts, tags, comments and other types of text-based content. A general form of an HTML element is therefore: <tag attribute1="value1" attribute2="value2">**content**</tag>. Some HTML elements are defined as empty elements and take the form <tag attribute1="value1" attribute2="value2" >. Empty elements may enclose no content, for instance, the BR tag or the inline IMG tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character "/", and that in empty elements the end tag is neither required nor allowed. If attributes are not assigned values, default values will be used in each case.

The main text of a web page typically occupies the center part of the web page, and this part of code contains dense text with few links or images. After preprocessing, we split the extracted text into a string sequence of N lines, denoted by an ordered set $L = \{p_1, p_2, ..., p_N\}$. Therefore, some spatial and content features in a text line can be readily used to differentiate text line importance.

As the spatial features are difficult to capture from the HTML source, in this paper, we only measure the spatial distance between a text line p_i, and the document's <body> tag in terms of the total number of lines in between, i.e., the index of p_i, in set L which is i. The content features in a text line could reflect the importance of this text line. Examples of content features include the length of the text line (measured in HTML bytes), the length of the output text line (the length of the content between tags, which is also measured in HTML bytes), its density (the ratio of the output text length to the text length), the number of links, and the number of images, etc. The following features are used to represent a text line.

{Index, TextLength, OutputTextLength, Density, LinkNum, ImgNum}

Take the first topical line in Fig. 1(b) as an example. The complete code of this line in the HTML source file is "And as the clock ticked down on the team’s time in the state, he gave the Nets a proper send-off: “Two minutes left to go in New Jersey, two!” ". After preprocessing, the distance between this line and the <body> tag in the original HTML source code is 287 (Index); the length (TextLength) of this code is 198 in which tags account for 28 bytes; therefore, the length of the output text/content is 170 (OutputTextLength); the Density of this topic line is the ratio of the output text length to the text

length, which is 0.86. Since there are no LINK tags or IMG tags in this line, the number of links and the number of images are both 0. Then, we can use a feature vector to represent this topic line, which is {278, 198, 170, 0.86, 0, 0}.

If the training data is composed of a number of HTML source files which are from different web sites, the spatial feature such as "Index" needs to be measured using the relative distance (normalized distance), i.e., i/N; where i is the original Index value and N is the total number of text lines in the document. The other features can be normalized in a similar way.

3.3 Potential topical line detection

To determine whether a text line belongs to the main text, many analytical approaches have been adopted so far. The two most commonly used techniques are statistical analysis and machine learning approaches. Here, we adopt a machine learning approach: association rule learning. Some classification approaches such as C4.5 decision tree (Quinlan 1986), back propagation neural networks (Lippmann 1987), and SVM (Cristianini and Shawe-Taylor 2000) can also be used for story line identification. We choose association rule learning for its simplicity and validity. Association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases (Hegland 2005). Some well-known algorithms are Apriori (Hegland 2005; Agrawal and Srikant 1994), Eclat (Zaki 2000) and FP-Growth (Han et al. 2004). They are suitable for discovering regularities in transaction data. However, the features used in our study are continuous data in general to which the above methods cannot be directly applied. A discretization process is needed, as detailed in Step 1 below.

In this paper, the fuzzy association rule mining approach in (Hegland 2005; Agrawal and Srikant 1994) is utilized to find interesting patterns from HTML source data. The raw feature values of some sample topical lines are shown in Table 2. They were manually extracted from a sample news web page, consisting of 13 topical lines and 6 items/features denoted as A-F, respectively.

Table 2 Thirteen topical lines in a web page HTML source. A: Index; B: TextLength; C: OutputTextLength; D: Density; E: LinkNum; F: ImgNum.

Topical lines	Items/Features
1	(A: 287) (B: 198) (C: 170) (D: 0.86) (E: 0) (F: 0)
2	(A: 288) (B: 861) (C: 482) (D: 0.56) (E: 2) (F: 0)
3	(A: 289) (B: 237) (C: 509) (D: 0.88) (E: 0) (F: 0)
3	(A: 289) (B: 237) (C: 509) (D: 0.88) (E: 0) (F: 0)
4	(A: 290) (B: 263) (C: 235) (D: 0.89) (E: 0) (F: 0)
5	(A: 291) (B: 650) (C: 449) (D: 0.69) (E: 1) (F: 0)
6	(A: 292) (B: 241) (C: 213) (D: 0.88) (E: 0) (F: 0)
7	(A: 293) (B: 504) (C: 301) (D: 0.60) (E: 1) (F: 0)
8	(A: 294) (B: 219) (C: 191) (D: 0.87) (E: 0) (F: 0)
9	(A: 295) (B: 287) (C: 259) (D: 0.90) (E: 0) (F: 0)
10	(A: 296) (B: 226) (C: 198) (D: 0.88) (E: 0) (F: 0)
11	(A: 297) (B: 391) (C: 363) (D: 0.93) (E: 0) (F: 0)
12	(A: 298) (B: 152) (C: 124) (D: 0.82) (E: 0) (F: 0)
13	(A: 299) (B: 211) (C: 156) (D: 0.74) (E: 0) (F: 0)

The fuzzy membership function for each item/feature is shown in Fig. 3, where MIN, Mean and MAX represent that item/feature's weighted minimum/mean/maximum values over its observed values in the topical region (tr) and that in the non-topical regions (non-tr), as defined by Equations (1), (2) and (3), respectively.

$$item.MIN = \alpha \cdot item.MIN^{tr} + (1-\alpha) \cdot item.MIN^{non-tr} \quad (1)$$

$$item.Mean = \alpha \cdot item.Mean^{tr} + (1-\alpha) \cdot item.Mean^{non-tr} \quad (2)$$

$$item.MAX = \alpha \cdot item.MAX^{tr} + (1-\alpha) \cdot item.MAX^{non-tr} \quad (3)$$

In the above equations, α is a weight value for balancing the importance between the topical region and the non-topical regions. Here, we set the value of α to 0.5. Take the item A (Index) as an example. The minimum value of A is 287 in the topical region and 1 in the other regions. Therefore, A.MIN is 144 according to Equation (1). The resulting MIN, Mean, and MAX values

of all the items/features for the web page in Fig. 1 are shown in Table 3. Now, the raw feature values can be represented by their probabilities of falling into three fuzzy regions - Low, Middle and High, respectively (see Fig. 3). Thus, three fuzzy membership values are produced for each item/feature in each line of text according to the predefined membership functions (see Fig. 3).

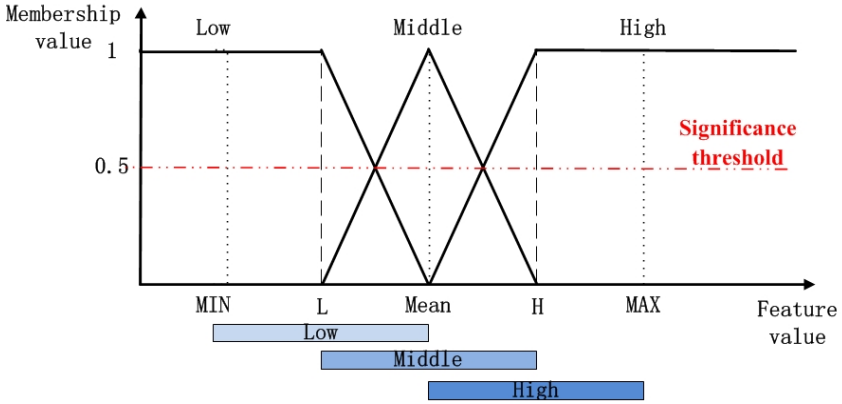


Fig. 3 The membership functions defined by MIN, Mean, and MAX values. L is the average of MIN and Mean, and H is the average of Mean and MAX.

Table 3.Some MIN/Mean/max feature values for the web page in Fig. 1

Items	MIN	Mean	MAX
A	144	308	473
B	81	210	1248
C	62	137	653
D	0.279	0.552	0.964
E	0	0	6
F	0	0	1

The proposed approach for finding fuzzy association rules is as follows:

Step1: Transform the original value v_{ij} of each item/feature I_j in the i^{th} topical line into a fuzzy set f_{ij} represented as $(f_{ij1}, f_{ij2}, ..., f_{ijh_j})$ using the given membership functions, where h_j is the number of fuzzy regions for I_j which is 3 in our case; f_{ijl} $(1 \le l \le h_j)$ is v_{ij} 's fuzzy membership value in the l^{th} fuzzy region R_{jl} of feature I_j. Take the feature A (Index) in the first topical line in Table 2 as an example. The raw value "287" of A is converted into a fuzzy set $(\frac{0.25}{A.Low}, \frac{0.75}{A.Middle})$ using the given membership functions in Fig. 3. This means that the probabilities of feature A falling into the three fuzzy regions (Low, Middle, and High) are 0.25, 0.75, and 0, respectively, in that text line. This conversion is repeated for each item/feature in each line, and the results are shown in Table 4.

Table 4 The transformed fuzzy REGION sets

Topical lines	Items/Features
1	$(\frac{0.26}{A.Low}, \frac{0.74}{A.Middle})(\frac{0.19}{B.Low}, \frac{0.81}{B.Middle})(\frac{0.87}{C.Middle}, \frac{0.13}{C.High})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
2	$(\frac{0.24}{A.Low}, \frac{0.76}{A.Middle})(\frac{1}{B.High})(\frac{1}{C.High})(\frac{0.96}{D.Middle}, \frac{0.04}{D.High})(\frac{0.33}{E.Middle}, \frac{0.67}{E.High})(\frac{1}{F.Low})$
3	$(\frac{0.23}{A.Low}, \frac{0.77}{A.Middle})(\frac{0.95}{B.Middle}, \frac{0.05}{B.High})(\frac{0.72}{C..Middle}, \frac{0.28}{C.High})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
4	$(\frac{0.22}{A.Low}+\frac{0.78}{A.Middle})(\frac{0.90}{B.Middle}+\frac{0.10}{B.High})(\frac{0.62}{C..Middle}+\frac{0.38}{C.High})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
5	$(\frac{0.21}{A.Low}+\frac{0.79}{A.Middle})(\frac{0.15}{B.Middle}+\frac{0.85}{B.High})(\frac{1}{C.High})(\frac{0.33}{D.Middle}+\frac{0.67}{D.High})(\frac{0.67}{E.Middle}+\frac{0.33}{E.High})(\frac{1}{F.Low})$
6	$(\frac{0.20}{A.Low}+\frac{0.80}{A.Middle})(\frac{0.94}{B.Middle}+\frac{0.06}{B.High})(\frac{0.71}{C..Middle}+\frac{0.29}{C.High})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
7	$(\frac{0.18}{A.Low}+\frac{0.82}{A.Middle})(\frac{0.43}{B.Middle}+\frac{0.57}{B.High})(\frac{0.36}{C..Middle}+\frac{0.64}{C.High})(\frac{0.78}{D.Middle}+\frac{0.22}{D.High})(\frac{0.67}{E.Middle}+\frac{0.33}{E.High})(\frac{1}{F.Low})$

8	$\left(\frac{0.17}{A.Low}+\frac{0.83}{A.Middle}\right)\left(\frac{0.98}{B.Middle}+\frac{0.02}{B.High}\right)\left(\frac{0.79}{C.Middle}+\frac{0.21}{C.High}\right)\left(\frac{1}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$
9	$\left(\frac{0.16}{A.Low}+\frac{0.84}{A.Middle}\right)\left(\frac{0.85}{B.Middle}+\frac{0.15}{B.High}\right)\left(\frac{0.53}{C.Middle}+\frac{0.47}{C.High}\right)\left(\frac{1}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$
10	$\left(\frac{0.15}{A.Low}+\frac{0.85}{A.Middle}\right)\left(\frac{0.97}{B.Middle}+\frac{0.03}{B.High}\right)\left(\frac{0.76}{C.Middle}+\frac{0.24}{C.High}\right)\left(\frac{1}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$
11	$\left(\frac{0.13}{A.Low},\frac{0.87}{A.Middle}\right)\left(\frac{0.65}{B.Middle},\frac{0.35}{B.High}\right)\left(\frac{0.12}{C.Middle},\frac{0.88}{C.High}\right)\left(\frac{1}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$
12	$\left(\frac{0.12}{A.Low},\frac{0.88}{A.Middle}\right)\left(\frac{0.90}{B.Low},\frac{0.10}{B.Middle}\right)\left(\frac{0.35}{C..Low},\frac{0.65}{C.Middle}\right)\left(\frac{1}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$
13	$\left(\frac{0.11}{A.Low},\frac{0.89}{A.Middle}\right)\left(\frac{1}{B.Middle}\right)\left(\frac{0.93}{C..Middle},\frac{0.07}{C.High}\right)\left(\frac{0.09}{D.Middle},\frac{0.91}{D.High}\right)\left(\frac{1}{E.Low}\right)\left(\frac{1}{F.Low}\right)$

Step2: Calculate the scalar cardinality of each fuzzy region R_{jl} in the transaction data as $count_{jl} = \sum_{i=1}^{n} f_{ijl}$ where n is the total number of topical lines. Take the fuzzy region A.Middle as an example. Its scalar cardinality is (0.74+0.76+…+0.89) which is 10.62. The scalar cardinality values of all the fuzzy regions calculated from the data in Table 5 are presented in Table 5.

Table 5 The scalar cardinality values of all the fuzzy regions

R_{jl}	$Count_{jl}$	R_{jl}	$Count_{jl}$	R_{jl}	$Count_{jl}$
A.Low	2.38	C.Low	0.35	E.Low	10
A.Middle	10.62	C.Middle	7.05	E.Middle	1.67
A.High	0	C.Middle	5.60	E.High	1.33
B.Low	1.09	D.Low	0	F.Low	13
B.Middle	8.73	D.Middle	2.16	F.Middle	0
B.High	3.18	D.High	10.84	F.High	0

Step3: Calculate $\max\_count_j = MAX_{l=1}^{h_j}(count_{jl})$ for each j in [1, m], where m is the number of items/features. Let max_R_j be the region with $\max\_count_j$ for item I_j. It will then be used to represent the fuzzy characteristics of item I_j in the subsequent mining process, and these fuzzy regions are group into a set L, namely the set of frequent fuzzy regions. Take item A as an example. Its *count* value is 2.71 for Low, 10.29 for Middle, and 0 for High. Since the *count* value for Middle is the highest among the three, the region Middle is thus used to represent item A in the later process. This step is repeated for every other item/feature. Thus, region Low is chosen for E and F, region Middle is chosen for B and C, and region High is chosen for D. The results of this step are shown in Table 6.

Table 6 The set of frequent fuzzy regions L

Frequent fuzzy regions	$Count_{jl}$
A.Middle	10.62
B.Middle	8.73
C.Middle	7.05
D.High	10.84
E.Low	10
F.Low	13

Step4: The fuzzy regions which are not in L are removed from the fuzzy region set for each topical line. Among the remaining fuzzy regions, we also remove those with f_{ijl} values below the significance threshold (0.5 as default). For example, since the membership values (f_{ijl}) of A.Middle, B.Middle, D.High, E.Low and F.Low are greater than 0.5, these fuzzy regions remain in the fuzzy region set for topical line 1, which will be further used to construct the fuzzy association rules. The final frequent fuzzy region set associated with each topical line is presented in Table 7.

Table 7 The topic lines with frequent fuzzy regions

Topical lines	Items/Features
1	$(\frac{0.74}{A.Middle})(\frac{0.81}{B.Middle})(\frac{0.87}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
2	$(\frac{0.76}{A.Middle})(\frac{1}{F.Low})$
3	$(\frac{0.77}{A.Middle})(\frac{0.95}{B.Middle})(\frac{0.72}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
4	$(\frac{0.78}{A.Middle})(\frac{0.90}{B.Middle})(\frac{0.62}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
5	$(\frac{0.79}{A.Middle})(\frac{0.67}{D.High})(\frac{1}{F.Low})$
6	$(\frac{0.80}{A.Middle})(\frac{0.94}{B.Middle})(\frac{0.71}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
7	$(\frac{0.82}{A.Middle})(\frac{1}{F.Low})$
8	$(\frac{0.83}{A.Middle})(\frac{0.98}{B.Middle})(\frac{0.79}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
9	$(\frac{0.84}{A.Middle})(\frac{0.85}{B.Middle})(\frac{0.53}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
10	$(\frac{0.85}{A.Middle})(\frac{0.97}{B.Middle})(\frac{0.76}{C.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
11	$(\frac{0.87}{A.Middle})(\frac{0.65}{B.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
12	$(\frac{0.88}{A.Middle})(\frac{1}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$
13	$(\frac{0.89}{A.Middle})(\frac{1}{B.Middle})(\frac{0.93}{C.Middle})(\frac{0.91}{D.High})(\frac{1}{E.Low})(\frac{1}{F.Low})$

Step5: Apply Apriori algorithm (Hegland 2005; Agrawal and Srikant 1994) to find fuzzy association rules for detecting topical lines. It starts with identifying the frequent single item sets. For example, the item A.Middle in Table 8 appears in all the 13 frequent region sets for the 13 topical lines, therefore, its support value is 13/13=1 which is greater than our default support threshold 0.5. Similarly, since B.Middle, C.Middle, D.High, E.Low and F.Low all have a support value greater than 0.5, they are each a frequent single item and are included into the frequent single item set L1 (see Table 8). The next step is to find all the frequent two-item sets (L2 in Table 8), using the same support value threshold. This process continues to discover higher-order item sets until no more frequent item set can be discovered.

Table 8 The fuzzy association rules discovered by applying Apriori algorithm (with a support threshold 0.5)

	Fuzzy association rules
L1	(A.Middle); (B.Middle); (C.Middle); (D.High); (E.Low); (F.Low)
L2	(A.Middle, B.Middle); (A.Middle, C.Middle); (A.Middle, D.High); (A.Middle, E.Low); (A.Middle, F.Low); (B.Middle, C.Middle); (B.Middle, D.High); (B.Middle, E.Low); (B.Middle, F.Low); (C.Middle, D.High); (C.Middle, E.Low); (C.Middle, F.Low); (D.High, E.Low); (D.High, F.Low); (E.Low, F.Low)
L3	(A.Middle, B.Middle, C.Middle); (A.Middle, B.Middle, D.High); (A.Middle, B.Middle, E.Low); (A.Middle, B.Middle, F.Low); (B.Middle, C.Middle, D.High); (B.Middle, C.Middle, E.Low); (B.Middle, C.Middle, F.Low); (B.Middle, D.High, E.Low); (B.Middle, D.High, F.Low); (D.High, E.Low, F.Low)
L4	(A.Middle, B.Middle, C.Middle, D.High); (A.Middle, B.Middle, C.Middle, E.Low); (A.Middle, B.Middle, C.Middle, F.Low); (A.Middle, B.Middle, D.High, E.Low); (A.Middle, B.Middle, D.High, F.Low); (A.Middle, B.Middle, E.Low, F.Low); (A.Middle, C.Middle, D.High, E.Low); (A.Middle, C.Middle, D.High, F.Low); (A.Middle, C.Middle, E.Low, F.Low); (A.Middle, D.High, E.Low, F.Low)
L5	(A.Middle, B.Middle, C.Middle, D.High, E.Low); (A.Middle, B.Middle, C.Middle, D.High, F.Low); (A.Middle, B.Middle, C.Middle, E.Low, F.Low); (A.Middle, B.Middle, D.High, E.Low, F.Low); (A.Middle, C.Middle, D.High, E.Low, F.Low); (B.Middle, C.Middle, D.High, E.Low, F.Low);
L6	**(A.Middle, B.Middle, C.Middle, D.High, E.Low, F.Low)**

In our example, the process terminates after discovering all the frequent 6-item sets (L6 in Table 8). Each L*i* in Table 8 thus represents a set of associated rules, delimited by semicolons, that

have exactly i items in each. Here, the association rule(s) with the most items will be adopted in our method. In our case, rule set L6 will be adopted to determine which HTML line is a potential topical line. The only one rule in L6 implies that if a HTML line 1) is located in the middle zone of the HTML code (A.Middle), 2) has medium text length and output text length (B.Middle, C.Middle), 3) has a relatively high text density (D.High), and 4) includes very few links or images (E.Low, F.Low), this line will be detected as a potential topic line. The reason we use only the highest order rule set is that they are the strictest in deciding which lines are topical lines, which has a very high precision but may introduce some false negatives. The sliding window technique that will be introduced in Section 3.4 can compensate for these false negatives to some extent.

3.4 Potential topical block segmentation

With the discovered fuzzy association rules, we can tell whether a given text line belongs to the topical (main text) region. This initial classification, though rough, labels most of the text lines correctly. However, if there are lengthy copyright notices, comments, and/or descriptions of other stories (not part of the main text), then those will likely to be labeled as topical lines too. Also, if there are descriptions around inline graphics that are part of some advertisement, or lengthy textual advertisements, these may also be labeled as topical lines. False negatives could also be observed when a topical line is not sufficiently long. To address these issues, we use two techniques. First, a sliding window technique is utilized to segment a web page to several potential topical blocks. The process is described below:

Step6: Associate the i^{th} text line with a Boolean variable M_i (TRUE represents that the line is a topical line and FALSE otherwise) according to the discovered fuzzy association rules in the previous step. Fig. 4(a) shows the content features of every text line in the example news web page from Fig. 1, and the labeling results according to Rule L6 are shown in Fig. 3(b).

Step7: Scan the entire HTML source file from top to bottom with a sliding window. The k^{th} potential topical block is represented as $B_k(start_k, end_k)$, where $start_k$ is the start position of the block, and end_k marks where it ends. $B_k(start_k, end_k)$ must satisfy the following conditions: 1) $M_i = FALSE$, if $start_k - \Phi \le i \le start_k$ or $end_k \le i \le end_k + \Phi$, where Φ is the length of the sliding window which is empirically set to 5 in our case, 2) $M_i = TURE$, if $i = start_k + 1$ or $i = end_k - 1$, and 3) $\underset{start_k \le i < j \le end_k}{Max} d_{ij} \le \Phi$, where both M_i and M_j are TRUE, and M_o is FALSE for $i < o < j$. In other words, in a topical block, no more than Φ-1 continuous non-topical lines can be included. This way, some false negatives from topical line detection can be tolerated. The detected topical blocks in the example news web page are shown in Fig. 4(c).

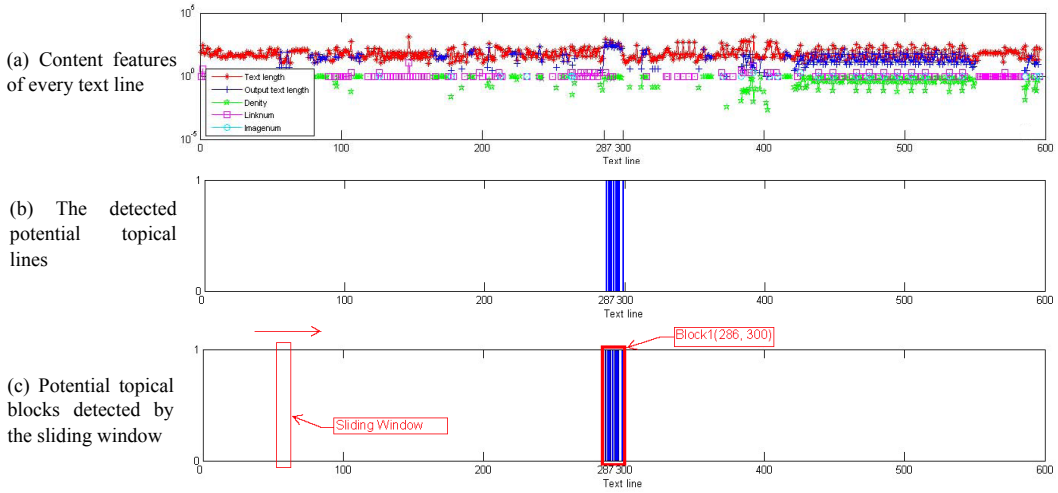


Fig. 4 Topical block detection for the example news web page in www.nytimes.com

3.5 Identifying the most informative blocks

After extracting potential topical blocks, the proposed FAR-SW algorithm identifies the most informative blocks. An informative block contains meaningful information that would be the

target of main text extraction. The other blocks that contain noise information such as advertisements, menus, or copyright statements are considered non-informative blocks.

As shown in Fig. 4(a), a topical line of a news page usually has a high density of texts with very few links or images, which motivates us to use a simple *Score* to measure the informativeness of every HTML line:

$$Score(p_i) = |T_i| + |O_i| + |D_i| - |L_i| - |I_i| \quad (4)$$

In calculating $|T_i|$, …, $|I_i|$, the normalization step mentioned in Section 3.2 is performed to ensure that all the resulted values are in the range of [0, 1]. Since the maximum score value is 3, a cutoff value of 1.5 is selected to indicate the informativeness of a HTML line. If $Score(p_i) > 1.5$, the line will be considered an informative line. To recognize informative blocks, we can use the average score of HTML lines in the block as a measure, as detailed in Equation 5:

$$Score(B_k) = \frac{\sum_{Start_k \le i \le End_k} p_i}{End_k - Start_k + 1} \quad (5)$$

Similarly, if $Score(B_k) > 1.5$, the block will be considered an informative block. As shown in Fig. 4(c), only one potential topical block is detected by Step 7 in this example. Since the informativeness score of the detected block is 1.5807 (greater than β), it is considered an informative block. The final extraction result contains the text lines from p_{286} to p_{300}, and the entire body of the true main texts (p_{287} to p_{299}) are highlighted by a red block, as shown in Fig. 1(a).

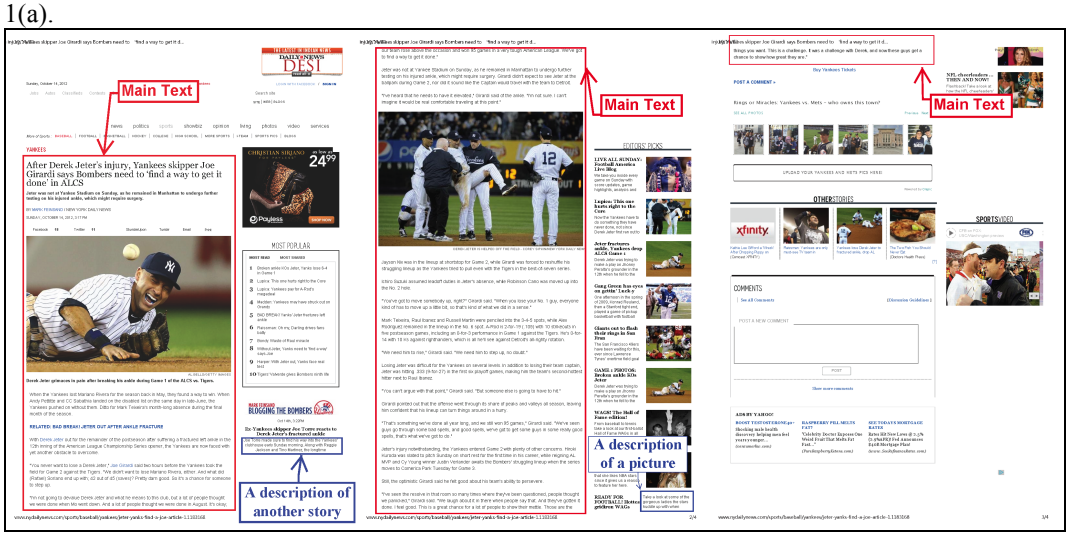


Fig. 5 An example of a News page in www.nydailynews.com including the main text and the descriptions of another story and picture

It is worth noting that this algorithm can help reduce false positives from topical line detection. A long line of advertising text and some descriptions of the other stories and pictures, which could be marked as topical lines in the previous step, are likely to be more 'isolated,' i.e., more likely to be surrounded by other non-topical text lines, than a typical topical line. Therefore, the *Score* would be relatively lower for any block that contains that line, when compared with the score of a typical topical block where true topical lines tend to stay closely together. As shown in Fig. 5, the news content is highlighted by a red-line polygon and the description of a story is highlighted by a blue-line polygon. Fig. 6(a) shows the content features of every text line in the example news web page from Fig. 5, and the labeling results according to the detected Fuzzy Association Rules are shown in Fig. 6(b). The marked potential topical lines include p_{507}, p_{509} - p_{527}, p_{636}, and p_{850}. Using a sliding window, we will get three potential topical blocks as shown in Fig. 6(c). Block1 (from p_{506} to p_{628}) contains some continuous topical lines; as its *Score* is 1.78, it will be considered as an informative block. The other two marked potential topical lines p_{636} and p_{850}, are those descriptions of advertisements or some other non-major stories (actually, they are false

positives). Notice that Equation 4 which is used to calculate the score of a block can also be used to calculate a score of a single text line In Block2 (from p_{635} to p_{637}), the *Score* of p_{636} is 1.67; although its value is higher than 1.5, the *Scores* of its surrounded lines (p_{635} and p_{637}) are 0.15 and 0.19; then, we will take the average of every line's *Score* as the final *Score* of this Block, which is 0.67. The *Score* of p_{850} is 1.55, its surrounded lines (p_{849} and p_{851}) are 0.28 and 0.33; the *Score* of Block 3 (from p_{849} to p_{851}) is 0.72. As the *Scores* of these two blocks are both lower than 1.5, they will not be considered as informative blocks. Therefore, the final extraction result contains the text lines from p_{506} to p_{528}, and the entire body of the true main texts (p_{507} to p_{527}) are highlighted by a red block, as shown in Fig. 5.

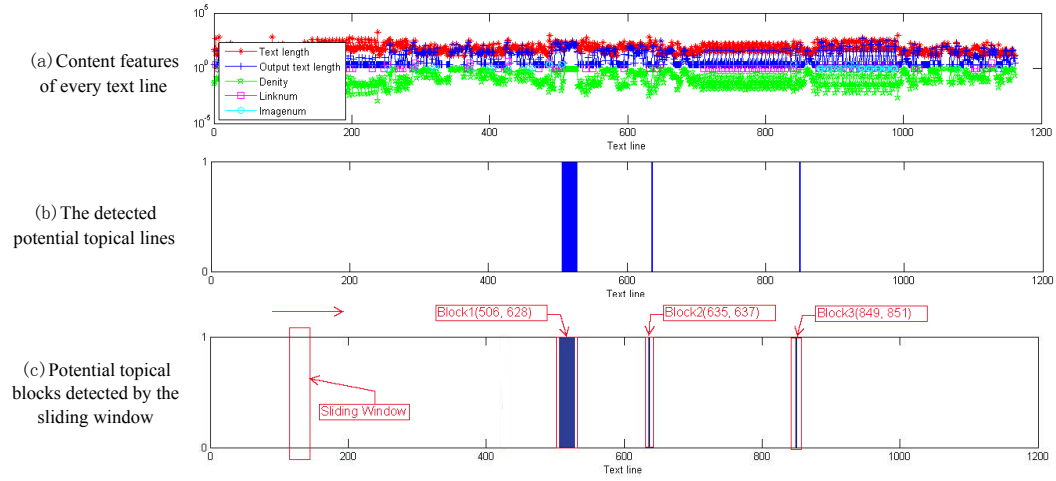


Fig. 6 Topical block detection for the example news web page in www.nydailynews.com

3.6 Evaluation Metrics

We adopt the metrics in (Eduardo 2009) to assess our results, which include precision, recall and F-measure. A higher value of precision indicates fewer wrong classifications, while a higher value of recall indicates less false negatives. They are calculated as follows:

$$\text{Precision} = \frac{|bag(C) \text{ I } bag(\text{Re}l(D))|}{|bag(C)|} \tag{6}$$

$$\text{Recall} = \frac{|bag(C) \text{ I } bag(\text{Re}l(D))|}{|bag(\text{Re}l(D))|} \tag{7}$$

Where $bag(C)$ denotes the bag of output text/content associated with a chunk of text C. $|bag(C)|$ is the length of output text/content (measured in HTML bytes) in C. $\text{Re}l(D)$ is the relevant content (main text) of a Web document D. It is common to use the harmonic mean of both measurements, called F-measure, such as the F1-*measure* defined by Equation (8) which weighs precision and recall equally important.

$$F1 - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{8}$$

Our first experiment uses the sample web page for both training and testing. The resulting precision, recall and F-measure values for testing this sample are 98.4%, 100%, and 99.2%, respectively, which are nearly perfect. This is an indication that our method is effective in removing noise, irrelevant information from web documents. More comprehensive experiments are presented in Section 4.

4 Experiments and analysis

In this section, we describe several experiments conducted on some real news Web sites in order to evaluate the performance of FAR-SW. Data sets used and employed fuzzy association rules are described in Section 4.1. We evaluate the performance by varying the length of sliding window in Section 4.2. The time complexity of our method is analyzed in Section 4.3. Finally, Section 4.4 provides the overall performance evaluation of FAR-SW.

4.1 Data Sets and fuzzy association rules

To evaluate the performance of the proposed approach, we built a test corpus of 1,600 news web pages collected from 16 websites, with 100 pages downloaded from each site. These data sets contain eight most popular Chinese sites and eight most popular English Web sites as described in Table 9. It would allow us to perform fair cross-validation tests to evaluate how well our models generalize the problem for unseen websites. A ten-fold cross-validation is performed in each subsequent experiment. The main text content of these news articles are manually labeled as the ground truth. We denote by "intra-site cross-validation" the experiments that train and test on pages from the same website. Similarly, we denote by "inter-site cross-validation" the experiments that train fuzzy associate rules on the overall sets of websites than the testing websites. As shown in Table 9, the average number of HTML lines (Avg. N) of every website is list in Column 3. The employed fuzzy association rules for each website are list in Column 4, from which, we can see there is no single universal fuzzy association rule for all of these 16 websites. The problems may lie in two aspects: 1) the main text of some websites (such as SOHO, QQ, NBC NEWS, WASHINGTON POST, LATIMES, and USA TODAY) is displayed in one HTML line, their fuzzy association rules often include B.High and C.High; 2) In some websites (such as IFENG, WASHINGTON POST, LATIMES, and USA TODAY), some words in main text are associated with hyperlinks, their fuzzy association rules often include E. Middle. To address these two problems, we change the membership functions of features B (TextLength), C (Outputtextlengh) and E (LinkNum) as shown in Fig. 7. Then, the fuzzy regions of I_B, I_C and I_E include two parts, which are Low' and High'. The membership functions for features A (Index), D (Density) and F (ImgNum) remain same as defined in Fig. 3. Column 5 of Table 9 lists the fuzzy association rule for all these 16 websites.

Table 9 Data sets and their fuzzy association rules

Site Abbr.	URL	Avg. N	FARs[1] (intra-site)	FARs[2] (inter-site)
SINA	news.sina.com.cn	1252	(A.Middle, **B. Middle, C. Middle**, D.High, **E.Low**, F.Low)	(A.Middle, B.High', C.High', D.High, E.Low', F.Low)
SOHO	news.sohu.com	1440	(A.Middle, **B.High, C. High**, D.High, **E. Low**, F.Low)	
163	news.163.com	1578	(A.Middle, **B.High, C. Middle**, D.High, **E.Low**, F.Low)	
IFENG	www.ifeng.com	1589	(A.Middle, **B.Middle, C.Middle**, D.High, **E. Middle**, F.Low)	
QQ	news.qq.com	2322	(A.Middle, **B.High, C.High**, D.High, **E. Low**, F.Low)	
XINHUA	www.xinhuanet.com	501	(A.Middle, **B.Middle, C.Middle**, D.High, **E.Low**, F.Low)	
PEOPLE	www.people.com.cn	476	(A.Middle, **B.Middle, C.Middle**, D.High, **E.Low**, F.Low)	
CHINA NEWS	www.chinanews.com	813	(A.Middle, **B. Middle, C.High**, D.High, **E.Low**, F.Low)	
YAHOO	news.yahoo.com	1906	(A.Middle, **B.High, C. Middle**, D.High, **E.Low**, F.Low)	
CNN	www.cnn.com	1531	(A.Middle, **B.Middle, C.Middle**,	

			D.High, **E.Low**, F.Low)
NBC NEWS	www.nbcnews.com	513	(A.Middle, **B.High,** **C.High**, D.High, **E. Low**, F.Low)
NYTIMES	www.nytimes.com	832	(A.Middle, **B.Middle, C.Middle**, D.High, **E.Low**, F.Low)
WASHINGTON POST	www.washingtonpost.com	3339	(A.Middle, **B.High,** **C.High**, D.High, **E. Middle**, F.Low)
LATIMES	www.latimes.com	6830	(A.Middle, **B.High,** **C.High**, D.High, **E. Middle**, F.Low)
FOX NEWS	www.foxnews.com	785	(A.Middle, **B. Middle, C. Middle**, D.High, **E.Low**, F.Low)
USA TODAY	www.usatoday.com	38	(A.Middle, **B.High,** **C.Middle**, D.High, **E.Middle**, F.Low)

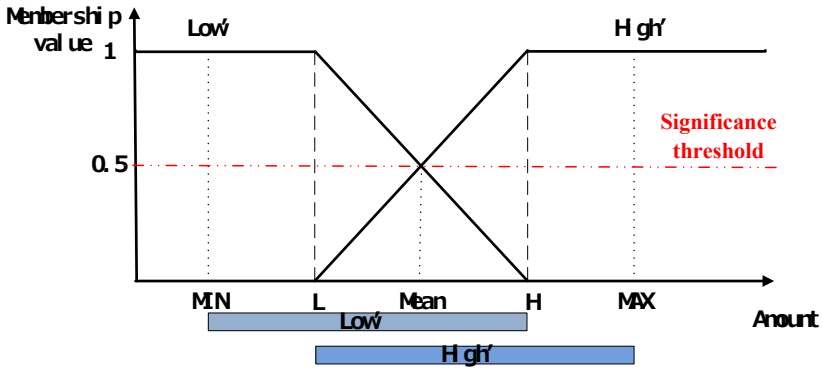


Fig. 7 The membership functions of features B, C and E defined by MIN, Mean, and MAX values. L is the average of MIN and Mean, and H is the average of Mean and MAX.

4.2 Variation with the length of sliding window

With the discovered fuzzy association rules, we can label most of the text lines as topical lines correctly. However, some false negatives could also be observed when a topical line is not sufficiently long. In Section 3.4, we use a sliding window technique to segment a web page to several potential topical blocks. In a topical block, no more than Φ-1 continuous non-topical lines can be included, where Φ is the length of the sliding window. This way, some false negatives from topical line detection can be tolerated. In fact, the accuracy of our approach changes with the length of sliding window. To prove this assertion, we apply the FAR-SW algorithm to webpages in IFENG, YAHOO, CNN and NYTIMES with the fuzzy association rules in Table 9. We use different Φ values to control the length of sliding window. If the length of sliding window (Φ) is set to 0, which means the sliding window will not work; we will treat every detected topic line as a potential topical block, in this case, some topical lines will be lost when their length is not long enough or they include some hyperlinks and images. Then, both Precision and Recall are very low as shown in Fig. 8(b,c). As the length of sliding window (Φ) grows, the number of potential topical blocks will decline rapidly; that means some false negatives from topical line detection can be tolerated, therefore, the extraction accuracy grows quickly. However, if we set Φ to a very high value, some lengthy copyright notices, comments, and/or descriptions of other stories will be also included in the detected main text; it will reduce the Precision of our approach. We need to set the length of sliding window to an appropriate value; in this paper, Φ is set to 5 through a larger number of experiments.

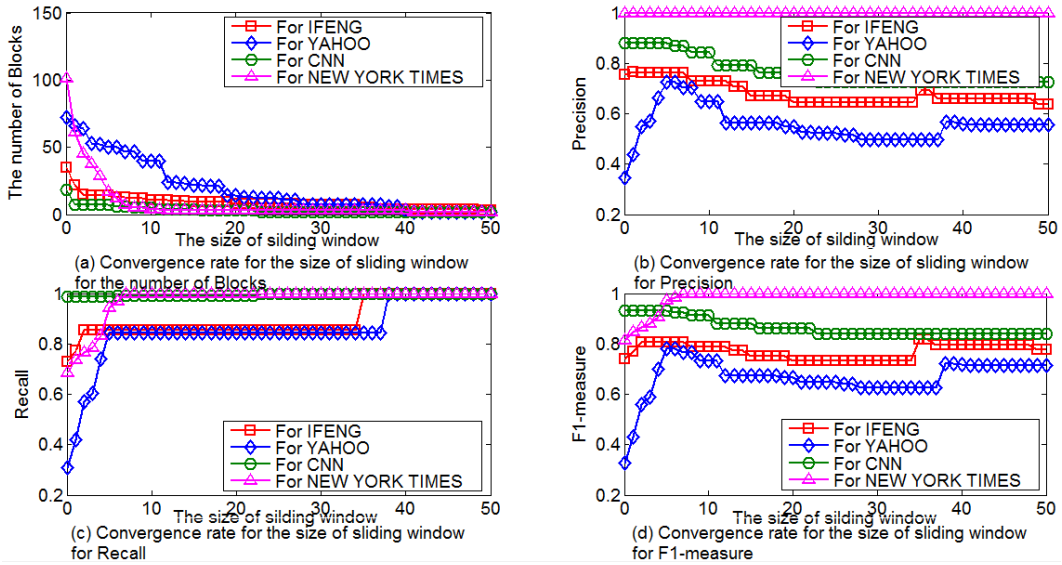


Fig. 8 The number of blocks, precision, recall and F1-measure changed with the size of sliding window.

4.3 Time complexity

We analyze the time complexity of our approach by considering the two major steps: the fuzzy association rules detection (Stage 1) and main text extraction (Stage 2). In Stage 1, the main influence on the time complexity comes from the Apriori algorithm, which is $O(m^2 n)$, where m is the number of items/features and n is the number of topical lines in database. In Stage 2, we need to render the web page to obtain the feature vector of every HTML line, it time complexity is $O(N)$, where N is the number of HTML lines in the given web page. Given the trained fuzzy association rules, the overall time complexity of our method is $O(N)$, which is linear time. For those DOM-based and Vision-based approaches, they need parse the HTML source into a DOM tree; the time complexity for this process is $O(N)$. In addition, both DOM-based and Vision-based methods involve some computationally expensive tasks, such as computing visual properties of HTML elements and pruning/merging DOM branches/sub-trees in DOM-based methods, and separator detection in Vision-based methods. The time complexity for these deep-level operations in DOM tree is $O(N^2)$. Our method is a statistics-based method, which does not need to take heavy and complicated searches in DOM tree and is expected to be faster than existing DOM-based and Vision-based approaches. This has been confirmed by our results.

4.4 Overall performance

The proposed FAR-SW extraction approach has been implemented in C++. All experiments were performed on a ThinkPad E420 laptop with two processors of Intel® Core i5-2410 @ 2.30GH and 2.00 GB RAM, running on Microsoft Windows 7 Home Bas. We also compare our method with existing DOM-based, Vision-based, and statistics-based approaches in efficiency and accuracy.

Table 10 shows the experimental results for the test sites. Columns 2 and 3 of Table 10 present, respectively, the average runtime and F1-measure attained by intra-site cross-validation for each site. Columns 3 and 4 present the results of inter-site cross-validation. The last six columns present the average runtime and F1-measure attained by WISDOM (DOM-based) (Kao et al. 2005), VIPS (Vision-based) (Kang et al. 2010) and WPMTE (Statistics-based) (Zhou 2009), respectively, for each site. It is worth noting that for VIPS, only intra-site cross-validation is performed because VIPS is tied to the design style of one particular site. For WISDOM and WPMTE, only inter-site cross-validation is performed, because typically inter-site testing accuracy is lower than that of intra-site testing due to the presence of a higher level of inconsistencies in the former, therefore, inter-site cross-validation can better assess the robustness of an algorithm and likely provide more insights in the meantime.

The average runtime for our method to extract news content from one news page is around 0.163 second, and about 0.055 second for statistics-based method. For those DOM-based and Vision-based approaches, the average runtime is much longer, about 1.300 and 1.027, respectively. Our method is faster than existing DOM-based and Vision-based approaches. In addition to efficiency, our approach can also achieve satisfactory accuracy. We observe that our FAR-SW based approach achieves the overall highest accuracy in inter-site cross-validation for all sites. The average inter-site accuracy of our approach is 90.3%, while WISDOM and WPMTE yield an average performance of 86.1% and 61.1%, respectively. As for their performance comparison on intra-site testing, VIPS yields the highest overall average F1 value (93.6%), but FAW-SW[1] also yields a very close performance (92.7%). Though VIPS performs slightly better than FAR-SW in intra-site cross-validation, it is site-based and thus less flexible, let alone the efficiency downside. WPMTE, like the other Statistics-based methods, is in general computationally efficient but yields significant worse performance (inter-site: 61.1%) than FAR-SW and WISDOM.

Table 10 The runtime and accuracy of the FAR-SW approach compared other existing approaches

Site Abbr.	FAR-SW[1] (intra-site)		FAR-SW[2] (inter-site)		WISDOM (inter-site)		VIPS(intra-site)		WPMTE (inter-site)	
	Avg. Time (s)	F1	Avg. Time (s)	F1	Avg. Time (s)	F1	Avg. Time (s)	F1	Avg. Time (s)	F1
SINA	0.181	91.6%	0.176	91.1%	0.655	80.5%	0.812	**92.3%**	**0.040**	58.6%
SOHO	0.148	90.7%	0.150	83.8%	1.158	78.4%	0.735	**91.2%**	**0.051**	52.5%
163	0.201	**96.1%**	0.198	**95.2%**	0.993	91.6%	1.013	94.2%	**0.061**	49.2%
IFENG	0.155	85.3%	0.159	74.6%	0.903	84.9%	0.617	**89.4%**	**0.058**	55.1%
QQ	0.199	**94.2%**	0.203	**93.8%**	1.047	83.2%	0.834	93.1%	**0.081**	62.3%
XINHUA	0.133	**93.1%**	0.127	**92.7%**	0.443	90.1%	0.382	89.9%	**0.025**	87.1%
PEOPLE	0.126	93.5%	0.134	91.5%	0.599	87.3%	0.789	**95.1%**	**0.029**	63.3%
CHINA NEWS	0.117	**97.9%**	0.119	**96.6%**	0.710	93.7%	0.609	94.3%	**0.039**	34.2%
YAHOO	0.222	80.3%	0.215	77.0%	4.171	**87.3%**	1.979	**87.1%**	**0.051**	45.6%
CNN	0.211	89.3%	0.212	81.6%	1.103	**83.1%**	1.036	**92.1%**	**0.033**	63.1%
NBC NEWS	0.127	90.9%	0.125	87.5%	1.429	92.1%	0.741	**98.4%**	**0.047**	47.8%
NYTIMES	0.134	**93.8%**	0.138	**93.4%**	0.588	87.7%	0.687	93.4%	**0.055**	81.4%
WASHINGTON POST	0.220	**97.2%**	0.195	**95.1%**	2.004	80.1%	1.503	94.1%	**0.047**	51.6%
LATIMES	0.237	93.7%	0.226	93.2%	2.787	76.3%	3.753	**94.5%**	**0.214**	43.2%
FOX NEWS	0.137	98.1%	0.133	97.8%	0.625	81.2%	0.693	**98.9%**	**0.029**	83.0%
USA TODAY	0.110	100%	0.096	100%	1.588	100%	0.248	100%	**0.019**	100%
Average	**0.166**	**92.7%**	**0.163**	**90.3%**	**1.300**	**86.1%**	**1.027**	**93.6%**	**0.055**	**61.1%**

5 Conclusions

In this paper, we have proposed a statistics-based approach to extract the main text content from web pages, which involves the following two steps 1) to identify potential topical text lines in the original HTML source, 2) to segment the web page into several potential topical blocks. In Step 1, the original HTML source is preprocessed and the text line features are extracted to represent every text line. Then, a supervised learning is performed to detect fuzzy association rules in training pages. In Step 2, the extracted fuzzy association rules from Step 1 are used to test whether each text line belongs to the main text. Next, a sliding window is applied to segment a web page into several potential topical blocks, based on the topical line detection results from the previous step. Finally, a simple selection algorithm is utilized to select the most informative blocks, which are then united to form the final detected topical region (main text).

We evaluate the efficiency and the accuracy of our proposed method by experimenting with real world data. The results show that the average runtime for our approach to extract news content from news pages is less than existing DOM-based and Vision-based approaches and comparable to other statistics-based methods. In addition to being efficient, our method can also achieve a satisfactory accuracy when compared with the others. In particular, our method yields the highest average inter-site cross-validation accuracy and the second best intra-site accuracy (second to

VIPS but very close). It is interesting to note that some image segmentation technologies can be used to divide the webpage into some visually independent sub-ranges. In most cases, the main text will be included in the maximum range. Then, we can use optical character recognition (OCR) technology to detect the textual information in the main text range. Using those recognized textual data, we can find the corresponding spatial distance in HTML source file. In this way, the detected spatial features in our method can be more accurate. We will combine this technology with the current approach in our future work.

Acknowledgments

This work was supported by JIANGSU INNOVATION PROGRAM FOR GRADUATE EDUCATION (Project NO: CXZZ12_0162) and THE FUNDAMENTAL RESEARCH FUNDS FOR THE CENTRAL UNIVERSITIES.

References

Gibson, D., Punera, K., Tomkins, A. (2005). The volume and evolution of web page templates. In Proceedings of the 14th international conference on WWW. 830-839, doi: 10.1145/1062745.1062763.

Theofanos, M.F., & Redish, J. (2003). Guidelines for Accessible and Usable Web Sites: Observing Users Who Work With Screen Readers. http://www.redish.net/content/papers/interactions.html/[Accessed 20 July 2008].

Koch, P.P. (2001). The Document Object Model: an Introduction. Digital Web Magazine. http://www.digital-web.com/articles/the_document_object_model/. [Accessed 10 January 2009].

Gupta, S., Kaiser, G., Neistadt, D., Grimm, P. (2003). DOM-based content extraction of HTML documents. In Proceedings of the 12th international conference on WWW. 207-214, doi: 10.1145/775152.775182

Kao, H.-Y., Ho, J.-M., Chen, M.-S. (2005). WISDOM: Web intrapage informative structure mining based on document object model. IEEE Transactions on Knowledge and Data Engineering, 17(5), 614-627.

Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y. (2003). Extracting content structure for web pages based on visual representation. In Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications. 406-417, ISBN:3-540-02354-2.

Kang, J., Yang, J., Choi, J. (2010). Repetition-based web page segmentation by detecting tag patterns for small-screen devices. IEEE Transactions on Consumer Electronics, 56(2), 980-986.

Alexjc. (2007). The easy way to extract useful text from arbitrary HTML. http://ai-depot.com/articles/the-easy-way-to-extract-useful-text-fromarbitrary-html/. [Accessed 5 April 2007].

Zhou, B., Xiong, Y., Liu, W. (2009). Efficient Web Page Main Text Extraction towards Online News Analysis. In Proceedings of the 2009 IEEE International Conference on e-Business Engineering. 37-41, doi: 10.1109/ICEBE.2009.15.

Quinlan, J.R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

Lippmann, R.P. (1987). An introduction to computing with neural nets. IEEE on ASSP Magazine, 4(2), 4-22.

Cristianini, N., Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines. Cambridge, UK: Cambridge University.

Hegland, M. (2005). The Apriori Algorithm - a Tutorial. In Mathematics and Computation in Imaging Science and Information Processing. Singapore: World Scientific Publishing Co. Pte. Ltd.

Agrawal, R., Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In Proceedings of the 20th International Conference on Very Large Data Bases. 487-499, ISBN:1-55860-153-8

Zaki, M.J. (2000). Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3), 372-390.

Han, J., Pei, J., Yin, Y., Mao, R. (2004). Mining frequent patterns without candidate generation. Data Mining and Knowledge Discovery, 8(1), 53-87.

Eduardo, S.L., Críston, P.S., Iam, V.J., Evelin, C.F.A., Eduardo, T.C., Raúl, P.R., Lúcio, C.T., Caio, D.V. (2009). A fast and simple method for extracting relevant content from news webpages. In Proceeding of CIKM, 1685-1688.