Multimedia Tools and Applications (2020) 79:11921–11945
https://doi.org/10.1007/s11042-019-08373-8

Web page classification: a survey of perspectives, gaps, and future directions

Mahdi Hashemi [1]

Received: 18 May 2018 / Revised: 21 August 2019 / Accepted: 9 October 2019 /
Published online: 10 January 2020

Abstract

The explosive growth of the amount of information on Internet has made Web page classification essential for Web information management, retrieval, and integration, Web page indexing, topic-specific Web crawling, topic-specific information extraction models, advertisement removal, filtering out unwanted, futile, or harmful contents, and parental control systems. Owing to the recent staggering growth of performance and memory space in computing machines, along with specialization of machine learning models for text and image classification, many researchers have begun to target the Web page classification problem. Yet, automatic Web page classification remains at its early stages because of its complexity, diversity of Web pages' contents (images of different sizes, text, hyperlinks, etc.), and its computational cost. This paper not only surveys the proposed methodologies in the literature, but also traces their evolution and portrays different perspectives toward this problem. Our study investigates the following: (a) metadata and contextual information surrounding the terms are mostly ignored in textual content classification, (b) the structure and distribution of text in HTML tags and hyperlinks are understudied in textual content classification, (c) measuring the effectives of features in distinguishing among Web page classes or measuring the contribution of each feature in the classification accuracy is a prominent research gap, (d) image classification methods rely heavily on computationally intensive and problem-specific analyses for feature extraction, (e) semi-supervised learning is understudied, despite its importance in Web page classification because of the massive amount of unlabeled Web pages and the high cost of labeling, (f) deep learning, convolutional and recurrent networks, and reinforcement learning remain underexplored but intriguing for Web page classification, and last but not least (g) developing a detailed testbed along with evaluation metrics and establishing standard benchmarks remain a gap in assessing Web page classifiers.

Keywords Web page classification · Image classification · Text classification · Deep learning · Machine learning · Artificial intelligence

✉ Mahdi Hashemi
mhashem2@gmu.edu

Extended author information available on the last page of the article

1 Introduction

The incredible growth of the World Wide Web (referred to as Web) has made it difficult to find Web pages that present satisfying information and to filter out the unwanted and harmful contents. On one hand, Web pages with offensive and harmful content, such as scam, phishing, violence, radicalism, cyber threats, porn, etc. have proliferated in the last decade. On the other hand, the vast amount of Web pages with diverse topics has hindered information retrieval and extraction models from providing optimal topic-relevant results.

Explosive performance and memory space growth in computing machines along with specialization of machine learning models for text and image classification have paved the way for automatic resolution of complicated semantic problems which seemed too unrealistic for any machine until a decade ago [31,106]. One of these problems is semantic classification of Web pages based on their content. Web page classification is the process of assigning a Web page to one or more predefined categories which plays a vital role in focused crawling, assisted development of Web directories, topic-specific Web link analysis, contextual advertising, and analysis of the Web's topical structure. The complexity and gravity of this problem along with the diversity of perspectives and models for Web page classification motivated us to provide a survey of the literature on this topic and shed light on its challenges and remained gaps. The rest of this paper is organized as follows. Section 2 presents a critical review of the state-of-the-art literature on Web page classification. This section investigates Web page classification models one-by-one under three main categories of text-based, image-based, and combined methods. Section 3 provides collective trends and insight into existing Web page classification models and points out research gaps, while Section 4 focuses on major limitations of Web page classification and offers potential doorways. Section 5 concludes the paper by providing a summary of research gaps and future directions.

2 A review of web page classification methods

Web page classification methods in the literature are explored in this section under the three categories of: (a) text-based, (b) image-based, and (c) combined text- and image-based classification. Categorizing automatic Web page classification methods in this way is motivated by the fundamental differences in how features are extracted from text and image for machine learning. However, categorization of Web page classification methods based on the applied classifier and accuracy is provided in Section 3. It is noteworthy that tokenization [44], non-alphanumeric character removal [86], stop-word removal [71, 86], lowercase conversion [86], and stemming [70] are pre-processing steps for all text-based classification methods.

2.1 Text-based classification

Human/sex trafficking or escort advertisements host and provide sexual services under categories of escort, adult entertainment, massage services, etc. Detecting such content is of interest to both criminal justice and companies. Law enforcement agencies would benefit from detecting such content due to its illegality. Companies are interested in detecting such content, so they can remove it to protect exposure to minors and report illegal activity on their platform. Thus, online pornography detection has gained substantial attention in the literature. Alvari

et al. [5] used semi-supervised SVM to detect such Web pages based on 12 binary features, extracted from the text. These binary features are:

- containing third person language,
- containing first person plural pronouns,
- containing high entropy in the content,
- containing three 4-g (resulting in three binary features),
- containing words and phrases of interest,
- the escorted individual being from countries of interest,
- multiple victims being advertised,
- low-weight victim,
- containing reference to infamous Web sites, and
- containing reference to spa massage therapies.

Ahmadi et al. [4] proposed to identify pornographic Web pages using a decision tree with meta-features, including:

- the number of pornographic words in the Web page,
- total number of words,
- ratio of pornographic words to all words,
- number of pornographic words in the Web page title,
- total number of words in the Web page title,
- number of hyperlinks,
- number of hyperlinks with a pornographic word in them,
- ratio of hyperlinks with a pornographic word in them to the total number of hyperlinks,
- number of blacklisted hyperlinks,
- number of images,
- number of images with a pornographic word in their name,
- number of videos,
- number of meta-tags containing a pornographic word,
- number of meta-tags with description,
- number of warning tags,
- number of warnings with a pornographic word in them,
- number of tooltips,
- number of tooltips with a pornographic word in them,
- number of frames, and
- number of colors.

However, classifying text mostly centers on the idea of counting the frequencies with which terms of a lexicon appear in the text to form a feature vector and applying those feature vectors to train a classifier. Kohonen self-organizing neural network [49, 50], nearest neighbor [19], Bayesian [32], naïve Bayesian [34, 53, 101], SVM [34, 83], and random forest [56] are among the applied Web page classifiers based on term-frequency feature vectors. Liparas et al. [56] applied the frequencies of 100 most frequent unigrams, 50 most frequent bigrams, 30 most frequent trigrams, and 15 most frequent fourgrams to classify the textual content of news article Web pages into four categories, using random forest. Li et al. [53] used naïve Bayesian to classify Web pages into ten different subjects based on term-frequency feature vectors

extracted from the title and main text. With the assumption of independent features in Naïve Bayesian classifier, each subject class will need a probability distribution function (PDF) for each term in the feature vector. They [53] bypassed the need to training samples (labeled Web pages) by developing these PDFs based on Wikipedia entity words instead. JingHua et al. [42] applied Dirichlet mixture distribution of term-frequency feature vectors to classify the textual content of academic Web pages into course, faculty, project, and student and to classify messages from 20 different news groups. They [42] took advantage of unlabeled samples in a semi-supervised learning based on expectation maximization algorithm and showed improvements in classification accuracy over the supervised model. A major problem with term-frequency feature vectors is the feature matrix's sparseness (too many zeros), especially for short texts, such as Tweets and review comments. Tian et al. [83] applied an information-gain-based feature selection approach to reduce the dimensionality of term-frequency feature vectors, extracted from part of a Web page, to 200. These feature vectors are later used to train an SVM for classifying 1400 short Web texts into seven categories: knife, crab, people, airplane, vessel, grapes, and gun. Li et al. [54] used an autoencoder, which reduces the dimensionality of term-frequency feature vectors through unsupervised learning, to initialize the synaptic weights of a deep neural network, which in turn classifies the emotion of social comments.

In addition to content information, a Web page contains structural information, i.e. tags or labels. In an attempt to consider the structural information in addition to textual information, Özel [68] and Lee et al. [51] considered the same term appearing in each group of the following tags as a different feature when constructing term-frequency feature vectors for Web pages:

- title <title>,
- header at level 1 < h1>, header at level 2 < h2>, header at level 3 < h3>,
- bold <b>, strong , emphasize , italic <i>,
- paragraph <p>,
- anchor <a>, and
- list item <li>,

arguing that: (a) their observations show that most of the important domain-specific terms appear under these tags and (b) similar tags have been used in earlier Web page classification studies that make use of HTML tags [45, 72, 84]. Terms that appear in only one document were eliminated, arguing that most of them are misspelled or irrelevant. One difference between [51, 68] is that the former took all the distinct terms from each of the above mentioned tags as features but the latter took select terms. Their experimental evaluation showed that tagged-terms as features increase the classification accuracy in comparison with using terms alone. Özel [68] proposed a genetic algorithm-based approach for binary classification of Web pages. A chromosome consists of a list of feature weights between 0 and 1. Each chromosome in the population is used as a Web page classifier. A threshold on the cosine similarity of the chromosome to each Web page's normalized term-frequency feature vector is used to classify that Web page. The chromosomes are evolved through genetic algorithms' mutation, cross-over, and selection operators based on their training classification accuracy as fitness. Three separate experiments on binary classification of Web pages into Computer Science related conference home pages or else, course home pages or else, and student home pages or else, showed that their [68] approach achieves higher accuracies than naïve Bayesian classifier in all

cases and kNN classifier in most cases. Lee et al. [51] applied simplified swarm optimization, a population-based evolutionary stochastic optimization technique, for binary classification of Web pages into art or not-art, computers or not-computers, health or not-health, and science or not-science. They [51] experimentally showed the higher accuracy of their approach in comparison with genetic algorithm, Bayesian classifier, and kNN.

An alternative to term-frequency (and term-presence) feature vector is term-TFIDF weight feature vector [74]. TFIDF weights, calculated from Eq. 1, intend to give higher weights to terms which appear in fewer documents and lower weights to terms occurring in many documents. This is achieved by multiplying a term's frequency by an inverse document frequency (IDF) factor. In Eq. 1, w_{ij} is the weight of the i-th term (t_i) in the j-th document (d_j), $TF(t_i, d_j)$ is the frequency of term t_i in document d_j, d is the number of documents, and $DF(t_i)$ is the number of documents containing the term t_i.

$$w_{ij} = TF\left(t_i, d_j\right) \times IDF(t_i) \tag{1}$$

$$IDF(t_i) = \log\left(\frac{d}{DF(t_i)}\right) \tag{2}$$

Term-TFIDF weight feature vectors are usually used for clustering purposes and their application to classification problems involves a subtle point. Let's assume a classifier is trained using term-TFIDF weight feature vectors of a corpus of training documents. If a new sample needs to be classified, the frequency of its terms need to be first multiplied by their corresponding $IDF(t_i)$, which were calculated based on the training corpus of documents. Selamat and Omatu [76] applied principle component analysis (PCA) to reduce the dimensionality of term-TFIDF weight feature vectors before inputting them into a neural network for classifying the textual content of Yahoo sports news Web pages into 12 categories. Fersini et al. [25] proposed an adjusted version of TFIDF weights, in Eq. 3, specifically customized for Web pages, where $TI(t_i, d_j)$ is the visual sensitivity of term t_i in document d_j.

$$w_{ij} = \left[TI\left(t_i, d_j\right) + TF\left(t_i, d_j\right)\right] \times IDF(t_i) \tag{3}$$

To calculate the visual sensitivity of terms, a Web page is first segmented into non-overlapping visual blocks that cover the entire Web page, based on the HTML DOM tree and horizontal and vertical lines that visually do not cross any blocks. Some visual blocks contain an image while others do not. An image-block is defined as a visual block which contains an image and its surrounding text. Visual sensitivity of terms which appear in no image-blocks is zero and the visual sensitivity would be zero for all terms if there are no images in the Web page. This is based on the assumption that images are the main consumer of the user's attention and terms that do not appear in any image-blocks do not attract any attention. Terms that are unequally distributed in image-blocks will be assigned higher visual sensitivities than terms that are equally distributed in all image-blocks.

A word's meaning is often contingent upon its surrounding words. A word might carry no subject-related burden by itself, but it can indicate the presence of a subject when combined with other words. On the other hand, the contextual information might wipe off the subject-related connotation from a subject-related word. Syntactic approaches, which tear the text apart into fixed-length term-frequency feature vectors, are incapable of effectively taking the contextual information surrounding the terms into account to discover the semantic or true

meaning of a text. Hu et al. [34] proposed that a term's frequency in the feature vector should be increased by one, only if it is closely surrounded by specific other terms that give a pornographic meaning to that term. However, this involves defining a massive number of analytical rules to determine in what precise contexts a term carries a specific meaning, making it more of a deductive approach than inductive.

Baecchi et al. [8] applied a rather complex approach to take into account the contextual information in classifying a Tweet's polarity (positive or negative). Their approach entails an unsupervised step to define a feature vector for every word and a supervised step to classify the polarity of a Tweet. The unsupervised step applies a large corpus of documents to produce a term-frequency feature vector for each document as rows in a matrix. The columns of this matrix would represent word feature vectors. Next, a modification step is applied to word vectors to move co-occurring words toward each other in the feature space. In the supervised learning step, the polarity of a Tweet would be the average polarity of its words. The polarity of each word in a Tweet is classified using logistic regression. The feature vector for the *i*-th word in a Tweet is produced based on its surrounding words. In other words, a fixed-length window is defined around the *i*-th word in a Tweet. The feature vector for each word in that window is fetched from the previous unsupervised step and the logistic regression takes all those feature vectors into account to classify the *i*-th word in that Tweet. Thus, the same word can be classified differently if used in different contexts. The problems associated with this approach are that: (a) the rationale behind moving co-occurring words toward each other in the feature space is not clear, it may never converge, is highly dependent on the text corpus, and is not clear how it contributes to the final goal of classifying Tweets, (b) a word's classification highly depends on the current position of its surrounding words in the feature space which dynamically changes during the unsupervised training, (c) a word itself does not play any role in its polarity classification because the window around the *i*-th word in a Tweet excludes the *i*-th word itself, in other words, the feature vector for the *i*-th word is constructed only based on its surrounding words, (d) the order of words in the Tweet is ignored during classification, and last but not least, (e) classifying a fixed-length set of words, with each word having a long and separate feature vector, is too ambitious of a task for logistic regression, not to mention it requires a tremendous training dataset considering how many combinations of a fixed-length set of words exist.

It is worth mentioning that we observed a common drawback in some studies where textual features, e.g. terms in term-frequency feature vectors, are hand-picked based on the entire dataset to best discriminate among the classes, before splitting them into training and test sets. This impugns the calculated cross-validation accuracy of the classifier due to the unfairness of the feature selection process based on the entire dataset.

2.2 Image-based classification

Extracting features from skin regions has been the most common approach for detecting the visual content of pornographic Web pages which demands elaborate analyses and intensive computations. Traditional pornographic image recognition techniques detect the skin color pixels in an image using color and texture information [4]. Since skin has a distribution with multiple peaks in the color space, Gaussian mixture models (GMMs) fitted to the skin color histogram have been a popular method for detecting skin pixels [43, 59, 82]. However, the expectation-maximization algorithm for finding the mixture model parameters has a high computational cost. In an effort to reduce this computational cost, Hu et al. [34] applied

adaptive bin sizes in the skin color histogram. Instead of GMM, Ahmadi et al. [4] used a neural network with one hidden layer to determine whether or not a pixel represents skin. The inputs to the neural network are the color features of the pixel and its four neighboring pixels.

After detecting skin pixels, Abin et al. [3] applied cellular learning automata on the skin probability map to bind skin pixels into skin regions. Instead of binding skin pixels to form skin regions (bottom-up approach), Xu et al. [93] extracted color and texture features from arbitrary-shaped segmented regions and used a GMM to decide which regions represent skin. In a similar but more sophisticated approach, Hu et al. [35] divided the image into regions by breaking the image into smaller windows as long as the color variance inside a window is larger than a threshold (top-down approach). Skin regions are then detected using color distribution information and relations between pixel values inside the region.

If skin regions resemble a human body shape, the image is labeled as pornographic [26, 37–39]. Due to the difficulty of considering all possible relative positions of body parts, some researchers attempted to extract features from segmented skin regions, instead of measuring the resemblance to a human body. The total area of the skin regions, the area of the largest connected skin region, the number of skin regions, and the number of colors in skin regions are the most common extracted features [6, 7, 11, 12, 55, 103]. Hu et al. [35] extracted 14 features to represent the global property of the image, 13 features from the largest skin region and its fit ellipse to represent the global human body property, and 4 features to represent human body's local properties (a total of 31 features). Some of these features, e.g. number of faces in the image, involve complicated and intensive computations. Ahmadi et al. [4] fitted one ellipse, called the global ellipse, to all skin regions and another ellipse, called the local ellipse, only to the largest skin region. They [4] extracted 29 global and local features, including the ratio of the local ellipse's minor axis to its major axis, normalized center of the local ellipse with respect to the image size, eccentricity of the local ellipse, difference between the angle of the major axis from the horizontal axis of the local ellipse and global ellipse, ratio of the largest skin region to the local ellipse's area, ratio of the skin region to the bounding box's area, ratio of the bounding box's width to its height, ratio of the largest skin region to all skin regions, seven normal moment invariants to describe shape features, and normalized edge direction histogram in 6 directions to describe textural features. In order to mitigate the misclassification of normal images rich in skin pixels, such as close range face images, as pornographic images, Hu et al. [34] extracted features from contours constructed from rectangular skin blocks, instead of extracting features directly from skin regions. Some researchers skipped the skin detection process and extracted other features, including Daubechies wavelets, normalized central moments, and color histogram [88, 89]. For example, bag-of-visual-words is a feature extraction method that discretizes an image into clusters of visually coherent regions referred to as visual words or interest regions. Ulges and Stahl [85] applied this feature extraction method for detecting child sexual abuse images. In their work, images are scaled to a width of 250 pixels. Each image is divided into overlapping rectangular patches of size 14×14 pixels at steps of 5 pixels. Each patch is described by applying the discrete cosine transform in YUV color space and selecting 78 low-frequency coefficients (36 for luminance and 21 for each chroma channel). Using this description, patches are clustered into 2000 clusters via k-means clustering. The size of clusters produces a feature vector of length 2000, which is used for binary classification.

The extracted features from different approaches are then used to train classifiers, e.g. nearest neighbor [34], kernel SVM [41, 73, 85], random forest [35], and neural network [4, 11, 43, 93, 103] to recognize pornographic images. Most pornographic image detectors suffer

from high false positive rates due to misclassifying normal images rich in skin or quasi-skin pixels.

Among Web page image recognition methods, for purposes other than pornography detection, Liparas et al. [56] applied five MPEG-7 visual descriptors, with a total of 320 features, in a random forest to classify news article Web images into four categories. The MPEG-7 standard specifies a set of descriptors, each capturing a different aspect of human perception, i.e. color, texture, and shape.

Content-based pornographic Web image detection historically has relied on hand-engineered visual features extracted from original images, e.g. skin color, texture, shape, and local invariant point. However, universal hand-engineered visual features are hard to select and design. For example, hand-engineered visual features are unstable due to various pornographic contents and skin color model cannot be accurate because of the skin color diversity and its sensitivity to changes in illumination and angle. Convolutional neural network (CNN), which does not rely on hand-crafted visual features, has recently outperformed other approaches in image classification [31, 46, 77, 80, 90, 98, 99] and object detection [22] due its independence from hand-crafted visual features and excellent abstract and semantic abilities [99]. CNNs make strong and mostly correct assumptions about the nature of images, namely, locality of pixel dependencies and stationarity of statistics. Therefore, in comparison with standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters which makes them easier to train. Nian et al. [66] applied the CaffeNet architecture [40], which is a slightly different version of AlexNet [46], for CNN to detect pornographic Web images in all styles. The network receives as input, raw 227×227 pixel RGB images and composes of five convolutional layers followed by three fully connected layers. Every convolutional layer is setup by four steps, which are sorted as convolution, ReLU, pooling, and normalization. The first two fully connected layers are composed of inner product, ReLU, and dropout while the last fully connected layer only has an inner product architecture. Finally, the softmax layer has a size equal to the number of target categories, labeled as pornographic and normal. To overcome the lack of a large number of training images, which is required for training CNNs, they [66] applied a data augmentation method which produces 10 times more training images. Their approach first scales the smallest side of all images to 227 pixels and then produces 10 additional training images out of each image by sliding a 227×227 window, 10 times, along each image. Additionally, to make the model invariant to changes in intensity, illumination color, and other transformations, augmented training images were produced via the following approaches: (a) translation and horizontal reflection, (b) Gaussian filter with different random variances, (c) random enhancements of RGB channels, and (d) adding random lighting noise. They also pre-trained their model using ImageNet dataset. To classify a test image, first its smallest side is scaled to 227 pixels. Then a 227×227 window is slid along the image with a stride of 5 pixels. The whole image is classified as pornographic if at least one of the patches is classified as pornographic. Many other studies have applied CNN for detecting pornographic images, e.g. Wang et al. [91], and videos, e.g. Perez et al. [69] and Moustafa [61]. They all achieve considerably higher classification accuracies than traditional approaches for image classification. The main drawbacks of CNN are its need for a large amount of training images, scaling to millions [46], and its requirement for equal-size input images [105].

Aforementioned image classification methods are content-based, i.e. they rely only on visual features. Context-based image classification, on the other hand, takes advantage of the text surrounding an image, if any, to fill the semantic gap between low-level visual features

and high-level semantic classification of images. Fauzi and Belkhatir [23] performed an experimental study to find the HTML tags and attributes that are more likely to contain textual information that is semantically relevant to Web images. Based on the authors' observation of 386 random Web images and 33 subjects' observation of 898 random Web images, the most probable elements to contain relevant textual information to Web images are ALT and SRC attributes of <IMG>, HREF and TITLE attributes of <A>, and strings enclosed by <A>, <TD>, <DIV>, and <*P*>. Based on their findings, Fauzi and Belkhatir [23, 24] applied a DOM Tree-based Web page segmentation algorithm that automatically segments Web pages into sections, with each section consisting of a Web image and its contextual information. The contextual information from these sections could be taken advantage of to enhance the classification accuracy of images beyond a mere content-based image classification. For example, Tian et al. [83] took advantage of term-frequency feature vectors, extracted from 200 to 300 Chinese characters around a Web image, to improve the classification accuracy of 1400 Web images into seven categories: knife, crab, people, airplane, vessel, grapes, and gun. However, the details about which part of the Web page these 200–300 characters are selected from are not clear. They reduced the dimensionality of term-frequency feature vectors to 200 using an information-gain-based feature selection method. An SVM classifier is trained using contextual features. A second SVM classifier is trained to classify an image based on its global visual features, including color, texture, and shape features. A third SVM classifier is trained to classify an image based on its local visual features. 256 color features are obtained by computing the hue entropy of non-overlapping blocks of 16×16 pixels in each image. Statistics extracted from the gray level co-occurrence matrix (GLCM) are used to quantify five common texture features: correlation, contrast, entropy, sum of squares, and inverse difference moment. The mean and variance of GLCM features are computed for four different angles: 0°, 45°, 90°, and 135° to obtain 10 rotation-invariance texture features. 27 shape features are extracted from three transition probability matrixes of hue saturation value (HSV) color space. 200 features are extracted from all aforementioned SIFT descriptors using k-means clustering. Local visual features include these 200 SIFT descriptors combined with 4200 local features, extracted from a three-level pyramid method. The posteriors of three classifiers (one contextual and two visual classifiers) are combined after assigning a weight to each one according to their classification accuracy.

2.3 Combined text- and image-based classification

In this section we review some attempts to combine visual and textual characteristics for Web page classification. All these combined approaches reported higher accuracies than individual image or text classifiers.

Fake Web sites garner illegitimate revenues by falsely posing as legitimate providers of information, goods, or services [15]. There are two types of fake Web site detectors: lookup systems and classifier systems. Lookup systems detect fake Web sites by comparing their URLs against a blacklist of known fake URLs, collaboratively reported by system users and online communities [52]. Lookup systems are fast in looking up URLs and have close to zero false positive rates but are reactive (because blacklists rely solely on reports of fake URLs by users which makes them slow to blacklist new fake Web sites) and thus, have high false negative rates [100]. Classifier systems detect fake Web sites based on features extracted from their content [100], body text, page style, images [57], image hashes, password encryption checks, URLs [14], and domain registration information, such as the domain name, host name,

host country, and registration date [52]. Classifier systems have lower overall accuracies in comparison with lookup systems [100] and are slower than lookup systems because classifying a Web site takes longer than looking up a URL in a blacklist. However, classifier systems are proactive and faster to blacklist new fake Web sites [1]. Hybrid systems combine classifier and lookup mechanisms by classifying only Web sites that are not already blacklisted [52]. For instance, Abbasi and Chen [2] proposed a hybrid fake Web site detector which applies an SVM classifier based on 6000 features extracted from body text (word- and character-level n-gram-frequency feature vectors), images (color frequencies arranged into 1000 bins as well as 40 image structure attributes, such as image height, width, file extension, and file size), linkages (e.g. number of inlinks and outlinks for each Web page), HTML tags (n-gram-frequency feature vectors), and URLs (token- and character-level n-gram-frequency feature vectors). In another effort to classify Web pages into phishing and not-phishing, Zhang et al. [101] defined a text classifier and a visual classifier. The text classifier uses the naïve Bayesian classifier, with a customized threshold, to classify term-frequency feature vectors, extracted from Web pages. The visual classifier transforms a Web page into a normalized fixed-size square image (100×100 pixels). Then, it classifies each image by putting a customized threshold on its visual similarity to its nearest neighbor in the phishing class, calculated based on the earth mover's distance method [86]. If the textual and visual classifiers disagree on the output class, they obey to the classifier that has a larger probability of correctness.

Liparas et al. [56] applied a random forest to classify the textual content and another random forest to classify the visual content of news article Web pages into four categories: Business-Finance, Lifestyle-Leisure, Science-Technology, and Sports. Weighted average is used to combine the posteriors of two random forests, where each random forest's weight is calculated based on the ratio of the inner-class to intra-class proximities for each class. The 195 textual features include the frequencies of 100 most frequent unigrams, 50 most frequent bigrams, 30 most frequent trigrams, and 15 most frequent fourgrams. The 320 visual features are extracted from five MPEG-7 visual descriptors of the biggest image in the Web page, including: color layout descriptor (18 features), color structure descriptor (32 features), scalable color descriptor (128 features), edge histogram descriptor (80 features), and homogeneous texture descriptor (62 features).

Baecchi et al. [8] attempted to consider both the text and the single image in classifying a Tweet's polarity (negative or positive). Each word in the text is classified individually and the overall Tweet's polarity is the average polarity of its words. After reducing the dimensionality of the color feature vector of the single image associated with a Tweet to 500 features, using a single-layer autoencoder, the shrunk feature vector of the image is concatenated with the feature vector of each word in the Tweet to classify its polarity using logistic regression. The major drawback here is that the image is overrepresented during the classification in comparison with the text, because the feature vector of the image is concatenated to the feature vector of every word in the Tweet. Besides, logistic regression is not flexible and sophisticated enough for classifying images based on their color feature vectors, let alone its concatenation with word feature vectors.

Following are some attempts to recognize pornographic Web pages. Hammami et al. [27] and Hammami and Chahir [28] combined an image classifier whose only input feature is the ratio of skin pixels to all pixels, with a text classifier which is trained using 20 features. Jones and Rehg [43] combined an image classifier, which works based on the resemblance between the skin region and a human body shape, with a text classifier using an OR operation. Hu et al. [34] used a naïve Bayesian classifier, which is trained using term-frequency feature vectors, to

calculate the probability of the textual content of a Web page being pornographic or not-pornographic. They [34] used a nearest neighbor classifier, which is trained using features extracted from contours that are constructed from rectangular skin blocks, to classify each image in the Web page. The probability that the visual content (combination of images) is pornographic is calculated as $(1\text{-}p_2)^{N1}(p_2)^{N2}$ and the probability that the visual content is not pornographic is calculated as $(p_1)^{N1}(1\text{-}p_1)^{N2}$, where N_1 is the number of images that are classified as pornographic, N_2 is the number of images that are classified as not-pornographic, $p_1 = 0.074$ is the false positive rate of their image classifier, and $p_2 = 0.027$ is its false negative rate. Eventually, the textual and visual posteriors are combined using multiplication.

Hu et al. [35] used a random forest classifier, which is trained using features extracted from skin regions that are constructed using a top-down approach, to classify each image in the Web page. To classify a video, it is first divided into a number of clips with small sizes (a few dozen frames). If at least one clip is classified as pornographic, the whole video is classified as pornographic. The posteriors of a clip belonging to either pornographic and not-pornographic classes are calculated by multiplying the posteriors of images inside it. The pornographic and non-pornographic posteriors for an image are obtained by multiplying the posteriors obtained from the image classifier by the posteriors obtained from the audio classifier. Recognition of adult sounds includes the extraction of 13 Mel-Frequency Cepstral Coefficients (MFCC) as audio features, from the audio section around the image, and their classification using a GMM. The term-frequency feature vector of a Web page's textual content is concatenated with the following features: the outputs of the random forest classifier for all images, the ratio of the number of pornographic images to the total number of images, and the ratio of the number of pornographic videos to the total number of videos. Such long feature vectors are used to classify test samples using kNN-based and SVM-based models.

Ahmadi et al. [4] applied a decision tree to meta-features of a Web page's textual content to obtain its probabilities of being and not-being pornographic. If the textual content is classified as pornographic or not-pornographic with a high confidence (large margin), the Web page is classified accordingly, regardless of the visual content. Otherwise, up to 5 images are randomly chosen from the Web page and classified using a neural network based on 29 local and global features extracted from skin regions. If the majority of images are classified as pornographic, the Web page is classified accordingly. Otherwise, the result of the textual content classification will be used to decide the final class.

3 Outline and research gaps

Table 1 provides a summary of Web page classification studies along with their applied features and dimensionality reduction method. Table 2 provides a summary of Web page classification studies along with their dataset, size of the dataset, number of classes, classification method, evaluation method, and classification accuracy. The studies in both tables are arranged in descending order of the classification accuracy. The classification accuracy in Table 2 is reported via F1 (also known as F-measure or F-score) which is the most common in the literature. We calculated F1 for studies where F1 was absent but the confusion matrix was present. In the absence of both F1 and confusion matrix in a study, the overall accuracy (OA) is reported. Figure 1. provides a scatter plot of the studies based on the applied classification method and the classification accuracy. If a method resulted in different classification accuracies in one study, only the highest accuracy is reported in Fig. 1. Figure 2 represents each

Table 1 Different Web page classification studies, their input features, and dimensionality reduction method (T: Textual content classification; CIT: Considering contextual information for text classification; H: Considering hyperlinks; SI: Considering structural information, i.e. HTML tags; M: Considering metadata; I: Content-based image classification; V: Video-based classification; CI: Context-based image classification; SS: Semi-supervised training)

Method	T	CIT	H	SI	M	I	V	CI	SS	Dimensionality reduction method
[101]	✓					✓				
[66]						✓				
[35]	✓					✓				
[69]							✓			
[5]	✓		✓		✓				✓	
[91]						✓				
[34]	✓	✓				✓				
[61]							✓			
[83]						✓		✓		Information-gain-based feature selection for contextual features
[53]	✓									
[76]	✓									PCA
[68]	✓			✓						
[4]	✓		✓		✓	✓				
[25]	✓	✓								
[56]	✓					✓				
[11]						✓				
[51]	✓			✓						
[85]						✓				
[42]	✓								✓	
[8]	✓	✓				✓				Autoencoder for visual features
[2]	✓				✓	✓				
[54]	✓									Autoencoder
[73]						✓				

classification method's frequency in the reviewed literature, alone or in combination with other classification methods.

The following general observations can be made from the above tables and figures. Web page classification has been mostly focused on pornographic Web pages until a decade ago. Ever since, the applications have diversified, beyond detecting pornographic Web pages, and textual features have become an integral component of Web page classification. According to Table 1, seven studies apply both images and text in classifying Web pages, ten ignore the images, and eight ignore the text. However, no specific patterns are observed on how the classification accuracy is affected by ignoring either the images or text.

Term-frequency has certainly been the most popular way of constructing fixed-length feature vectors for the textual content, while the application of metadata, structural information, and contextual information surrounding the terms are new, rare, and underexplored. Application of contextual information for text classification goes back to over a decade ago but has rarely been the focus of attention. Metadata have rarely played a role in Web page classification because structural information has appeared to be more effective in Web page classification. Despite context-based image classification has been applied in one image-based Web page classification, it might not be as effective in combined Web page classification, because textual features are already considered in a separate classifier.

Investigating the Web pages that hyperlinks refer to, can be helpful in Web page classification, but has only been considered in two studies. Hyperlinks could be applied to improve the classification accuracy. In case the Web page classifier is highly uncertain about a specific

Table 2 Different Web page classification studies, along with their dataset and classification method, arranged in descending order of the classification accuracy

Method	Dataset	Number of samples	Number of classes	Classification method	Evaluation method	Accuracy%
[101]	Phishing and non-fishing Web pages	~9000	2	• Naïve Bayes for text classification • Nearest neighbor for image classification	One-fold	F1 = 99~100
[66]	Pornographic and non-pornographic Web pages	48,600	2	• CNN	One-fold	OA = 99
[35]	Pornographic and non-pornographic Web pages	3090	2	• Random forest for image classification • kNN-based and SVM-based methods for text classification	One-fold	F1 = 98~99
[69]	Pornographic and non-pornographic Web pages	1000	2	• CNN	Two-fold	OA = 98
[5]	Human trafficking and non-human trafficking Web pages	20,000	2	• SVM	Ten-fold	F1 = 95
[91]	Pornographic and non-pornographic Web pages	150,000	3	• CNN	One-fold	F1 = 95
[34]	Pornographic and non-pornographic Web pages	~4000	3	• Naïve Bayes for text classification • Nearest neighbor for image classification	One-fold	F1 = 94
[61]	Pornographic and non-pornographic Web pages	800	2	• CNN	Five-fold	OA = 94
[83]	Web page categories of knife, crab, people, airplane, vessel, grapes, and gun	1400	7	• SVM	Three-fold	F1 = 92
[53]	Web page categories of culture, education, entertainment, finance, health, religion, government, science, sport, and travel	17,431 18,099 19,444	10	• Naïve Bayes	One-fold	OA = 93 OA = 92 OA = 91
[76]	Yahoo sports news Web page categories	4096	12	• Neural network	One-fold	F1 = 90
[68]	Course-related and not related Web pages	1051	2	• Genetic algorithm	One-fold	F1 = 91
	Conference and non-conference Web pages	292				F1 = 90
	Student and non-student Web pages	5405				F1 = 69
[4]	Pornographic and non-pornographic Web pages	1585	3	• Decision tree for text classification • Neural network for image classification	One-fold	F1 = 87
[25]	Yahoo Web page categories	10,000	5	• Multinomial Naive Bayes • SVM • 1 nearest neighbor • 5 nearest neighbors • Naive Bayes • Decision Tree • Bayesian Network	Ten-fold	F1 = 89 F1 = 89 F1 = 87 F1 = 87 F1 = 85 F1 = 81 F1 = 81
[56]	News article Web page categories	1043	4	• Random forest for text classification	One-fold	F1 = 85

Table 2 (continued)

Method	Dataset	Number of samples	Number of classes	Classification method	Evaluation method	Accuracy%
				• Random forest for image classification		
[11]	Pornographic and non-pornographic Web pages	11,005	2	• Neural network • k-nearest neighbors • SVM • Generalized linear model	Ten-fold	F1 = 83 F1 = 82 F1 = 82 F1 = 73
[51]	Health-related and not related Web pages	~1000	2	• Simplified swarm optimization	One-fold	F1 = 81
	Computer-related and not related Web pages	~1000				F1 = 78
	Science-related and not related Web pages	~1000				F1 = 78
	Art-related and not related Web pages	~1000				F1 = 73
[85]	Child-pornographic and non-child-pornographic Web pages	2000	2	• SVM	Five-fold	OA = 79
[42]	Academic Web page categories	4199	4	• Dirichlet mixture distribution	One-fold	F1 = 77
	Newsgroup Web page categories	19,946	20			F1 = 77
[8]	Tweet polarities (Sanders Corpus dataset)	3625	2	• Logistic regression (textual and visual features are concatenated)	Ten-fold	F1 = 73
	Tweet polarities (Sentiment140 dataset)	1,700,000			One-fold	F1 = 83
	Tweet polarities (SemEval-2013 dataset)	13,434			One-fold	F1 = 73
	Tweet polarities (SentiBank Twitter dataset)	6,03			Five-fold	F1 = 57
[2]	Fake and non-fake Web pages	1400	2	• SVM (textual and visual features are concatenated)	One-fold	F1 = 71
[54]	Social comment emotions (SinaNews dataset)	1246	6	• Deep neural network	One-fold	F1 = 60
	Social comment emotions (ISEAR dataset)	7666	7			F1 = 51
	Social comment emotions (SemEval-2007 dataset)	4570	8			F1 = 39
[73]	Pornographic and non-pornographic Web pages	69,260	2	• SVM	One-fold	F1 = 19

Web page's class, the Web pages pointed to by hyperlinks could be classified and their result could be incorporated in classifying the original Web page. More sophisticated approaches could be developed to recognize and discard videos, images, texts, and hyperlinks that are inconsistent with a Web page's theme.

With the explosive growth of unlabeled Web data and diversity of features that could be extracted, semi-supervised training and dimensionality reduction methods have also gained more attention in recent studies. However, we could not find any study that measures the effectiveness of features in distinguishing among Web page classes, or in other words measures the contribution of each feature in the classification accuracy. This is an important task, in order to filter out correlated features or features that make little to no contribution in distinguishing among classes [29, 30].

According to Table 2, best accuracies are achieved when kNN or naïve Bayes are applied, alone or in combination with other classifiers. Random forest and decision

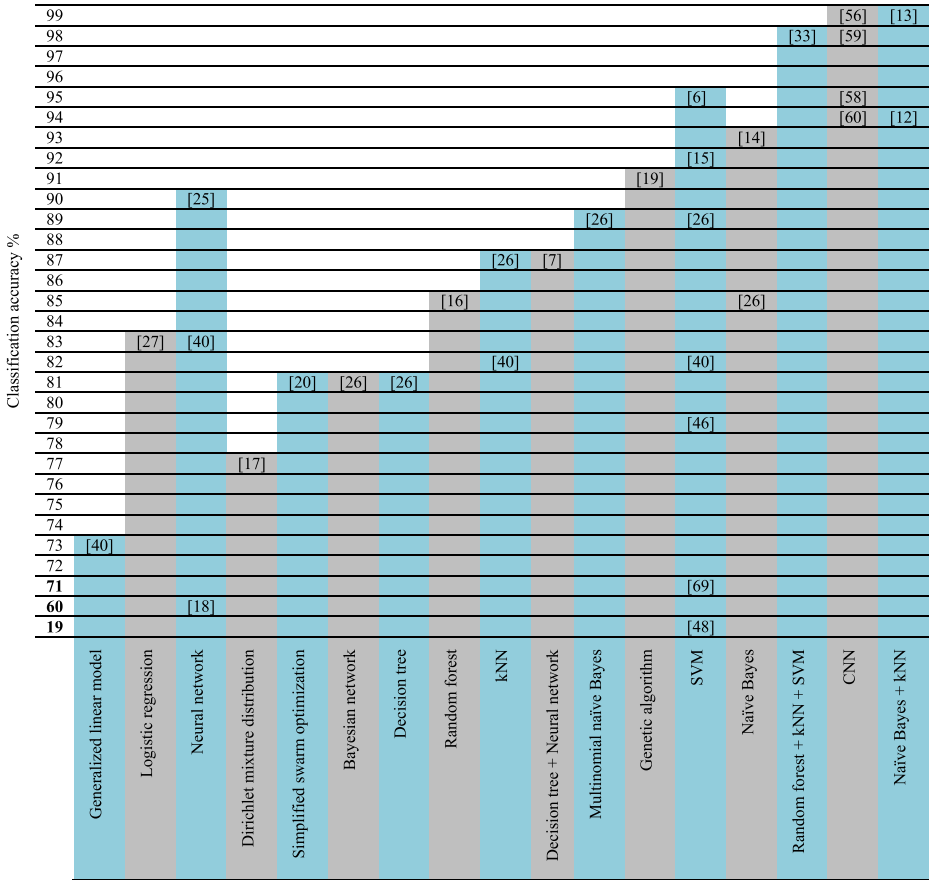

Fig. 1 Different Web page classifiers and their classification accuracy

tree are the following most accurate classifiers. SVM, the most frequently applied method, achieves a wide range of accuracies from 19% to 95% in different studies. The CNN model in Fig. 1 achieves the highest accuracies in image classification. SVM is the most frequently applied classifier according to Fig. 2, followed by neural network, kNN, naïve Bayes, random forest, and decision tree. Reinforcement learning remains as one of the approaches that has yet to be applied to Web page classification.

4 Limitations and potential doorways

This section underscores some of the limitations in front of automatic Web page classification. It also presents solutions that have been used to overcome similar challenges in the literature. The studies reviewed in this section are not mentioned in tables and figures in Section 3 because they do not specifically focus on Web page classification.

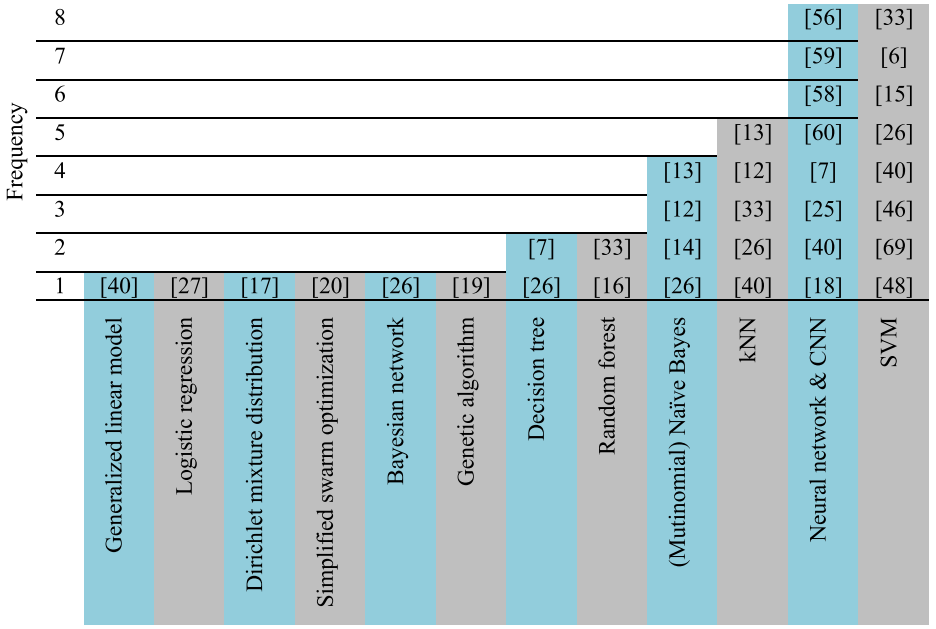

Fig. 2 Each classification method's frequency in the reviewed literature, alone or in combination with other classification methods

4.1 Limited training data

The limited number of labeled Web pages poses a challenge to effective training of automatic Web page classifiers. Finding a way to engage unlabeled Web pages in a semi-supervised classification process would be a potential solution to this problem. Also, representing Web pages in different feature spaces, if possible, would multiply the labeled samples. Fakeri-Tabrizi et al. [21] applied both these solutions to overcome the lack of sufficient labeled images when training their classifier. They proposed a self-learning approach which takes advantage of multiview representations to produce pseudo-labeled training data when a large number of unlabeled samples are available.

Before explaining their proposed methodology, we need to briefly describe the two concepts of multiview-learning and self-learning. Some observations can be represented in several feature spaces. These may be naturally different views on the same object, such as poses in object recognition, artificially generated views, such as different language versions of the same document, or different sets of features obtained from different means, such as color-based features vs. descriptive text for an image. Multiview learning trains a separate predictor over each view (called view-specific predictors) and combines them in order to improve the overall performance beyond that of predictors trained on each view separately. Self-learning (or self-training) is a semi-supervised learning technique which adds the confidently classified unseen samples to the training dataset.

Fakeri-Tabrizi et al. [21] trained different classifiers in different views. An unlabeled sample is classified using each view-specific classifier. If at least half of the classifiers agree on the same label, each having a confidence, obtained based on the distance between the sample and the class boundary, higher than a threshold, determined based on the upper bound of the

transductive Bayes error, that sample is pseudo-labeled and added to the training set. The view-specific classifiers are then re-trained and this procedure is repeated until no unlabeled sample could be added to the training set. Two major weaknesses of multiview self-learning are susceptibility to dependency (or correlation) between views and biasing the model with its own prediction mistakes.

Despite Fakeri-Tabrizi et al. [21] showed the effectiveness of their approach for image classification using SVM view-specific classifiers (with six visual views obtained from the image and one contextual view obtained from the user tags associated to each image), their approach could be extended to Web page classification. From a more conventional perspective in handling multiple views, Yang et al. [96] proposed to combine all the features from different media, including text and image, using an autoencoder, which also reduces the dimensionality, and apply the resulting feature set for unified classification through a single classifier.

4.2 Overlooking structural information

A Web page is a tree where each node represents a structural element. A node contains two types of information: tag or label information and content information. For example, for a node with the label paragraph, the content will be the paragraph text. While Web page classification has mostly focused on flat content classification, it has not paid much attention to the structural information. How to efficiently apply the structural information in Web page classification is yet to be fully investigated.

To find potential solutions to this challenge we allude to another field of research which concentrates on classifying structured documents (i.e. XML) using their structural information. In the simplest form, they train a separate classifier for each document element [20, 95]. These base classifiers are then combined for classifying the whole document. In a more sophisticated form, they combine term-frequencies and document elements in one feature vector [17, 18, 97, 102]. Denoyer and Gallinari [17], Yi and Sundaresan [97], and Diligenti et al. [18] applied the Bayesian probability to calculate the posteriors based on both textual content and structural information. In other words, the Bayesian probability comprises two parts: one for the structural probability and another for the content probability and the document parts are classified in the context of the whole document. Zhao et al. [102] applied extreme learning machine to classify structured documents, after incorporating structural information into term-frequency feature vectors.

4.3 Limitations in visual feature extraction and their classification accuracy

Low accuracy and extracting complicated hand-crafted features are two challenges in front of classifying the visual content of Web pages. CNN has been shown to surpass other image classification approaches in certain applications [22, 46, 77, 90, 98] because of its independence from hand-crafted visual features and excellent abstract and semantic abilities. Yet, CNN has not been adequately applied in Web page classification, mainly because of two reasons: it needs a large amount of training images, up to millions [46], and it receives only equal-size images as input [105]. Pre-training [66, 67] and data augmentation via translation, horizontal reflection, and altering the intensities of RGB channels [46, 66] have been widely proposed to overcome the limited number of training images.

Scaling [33, 46, 66, 67, 75, 78], cropping only the central patch of the image [75], or cropping several patches through sliding a window over the image [33, 66, 67] have been

dominant approaches for producing equal-size inputs for CNN. For example, Krizhevsky et al. [46] first rescaled all images such that their shorter side would be of length 256, and then cropped out the central 256 × 256 patch. Sliding a window across an image to produce equal-size image patches carries the risk of mislabeling some patches because they miss the specific pattern that exists in the image. For example, a training image might have a specific label because a small symbol is present at the corner or center of the image. All patches extracted using the sliding window will have the same label as the original image while not all of them contain that symbol. Scaling all images to a fixed size using interpolation carries the risk of deforming the specific pattern that the CNN is looking for to label an image. To prevent this deformation when scaling smaller images up to the fixed size, zero-padding was proposed as an alternative to interpolation [105].

4.4 Overlooking the sequence of terms in the textual content

Text is inherently sequential since one's comprehension of previous words will help his comprehension of subsequent words. The sequence of terms in the textual content of Web pages has been overlooked in the existing literature on Web page classification. As mentioned before, term-frequency feature vectors lose the structure of the text by tearing it apart into terms and their frequencies. Despite some researchers have proposed modifications of term-frequencies to partly take the contextual information around terms into account, their scope in considering the surrounding terms is very limited and they are not easily extendable to large datasets. A possible solution to this challenge is recurrent neural network (RNN) which provides the possibility of classifying a text as a whole while taking both the sequence of a stream of textual information of arbitrary length and their contextual information into account.

Another major weakness of term-frequency feature vectors, yet to be addressed, is their inability to distinguish among documents which are concerned with the same subject and use the specific words belonging to that subject but belong to different categories. For instance, pornographic text detectors have difficulty in correctly classifying some border documents on medical, sex education, sexual rights, and sports [4]. RNN offers a solution to this challenge as well, because it captures more contextual and structural details from the text than just term frequencies, enabling it to distinguish among documents that a classifier based on term frequencies might not.

Because of its capability to overcome these challenges, we elaborate more on the proper application of RNN to text classification. It has been highly emphasized in the literature that RNN is capable of classifying sequences of different lengths (e.g. messages and documents of arbitrary lengths), but so are conventional approaches based on term-frequency feature vectors. It also needs to be clarified that despite RNN is capable of classifying a sequence of tokens with arbitrary length (as are other conventional text classification methods), the actual feature vectors that are inputted into the RNN must be of the same length, (as they must for other conventional text classification methods).

RNN has been used for named entity recognition [13, 16, 36, 47], relation classification [58, 94, 104], language modeling [60], machine translation [9, 63, 79, 81], question answering [62–64, 87], and text classification [48, 63–65]. Although technical and graphical explanations of RNN are abundant in the literature, here we provide a conceptual explanation of how an RNN classifies a textual document, what it receives as input, how it considers the sequence of terms, and how it is capable of handling texts of different lengths. The term document, used in the following algorithm, could refer to a single word, a sentence, or a group of sentences and paragraphs, as long as its definition is consistent across the algorithm. The classification steps by RNN are as follow:

RNN steps for text classification	
1. Tokenization	The document is broken into tokens, which could be characters, terms, combination of terms, or sentences. It is up to the user to define the token and break the text into tokens. Conventionally, individual terms are considered as tokens.
2. Stop-word removal	It is up to the user to decide whether or not to keep all the tokens or dispose of unnecessary tokens, such as stop words. It can be argued that in a semantic sequence, stop words may be an indication of a unique sequence that will influence its classification and not removed.
3. Stemming	It is up to the user to decide whether or not to stem the terms, so terms from the same root will be considered as the same token.
4. Constructing a feature vector for each token	We need to create a feature vector for each token. **These feature vectors must be all of the same length.** It is up to the user to decide if the feature vectors must be unique or for any reason some tokens can have the same feature vector. It is up to the user to decide how to construct these feature vectors. Word embeddings are conventionally used as term feature vectors.
5. Classify the first token	The RNN receives the feature vector for the first token in the document and generates a class label as output. However, we do not consider this as the class label for the entire document yet.
6. Classify the next tokens, one by one, in the same order, until the last token in the document.	The RNN receives the feature vector for the second token in the document. It produces a class label as output. There are two important points here: • **The tokens must be fed to the RNN in the same order that they appear in the document.** • When the RNN attempts to classify the second token, it has in its memory how and why it assigned a specific class label to the first token and it applies that knowledge when it attempts to classify the second token. In other words, **while the first token is classified independently, the second token is classified not only based on its own feature vector, but also based on how and why the previous token was classified in a specific category**. **This is why RNN claims that it takes the sequential and contextual information into account.** Then the RNN receives the feature vector for the third token in the document. When it attempts to classify this feature vector, it also remembers and applies the knowledge of how it classified all the previous tokens. All tokens need to be fed to the RNN, one by one, until the last token. **This is why RNN is said to be capable of classifying sequences of arbitrary length.** There are two general strategies to determine the class label of the whole document: **the class label assigned to the last token or the most frequent class among all the tokens.**

If a second document needs to be classified, the RNN first wipes its memory of how it classified the tokens from the previous document, and then follows steps 5 and 6 from the above algorithm. In other words, the RNN classifies the first token of the second document independently, regardless of the previous document's classification. For example, if document is defined as a sentence, the user

Table 3 Publically available datasets for Web page classification

Name of the dataset	Source	Used in
Open Directory Project Web site	http://www.dmoz.org	[51, 68]
WebKB	http://www.cs.cmu.edu/~webkb	[42, 68]
20NEWSGROUPS	http://qwone.com/~jason/20Newsgroups	[42]
Flickr photo stock	https://www.flickr.com	[85]
Corel	https://archive.ics.uci.edu/ml/datasets/corel+image+features	[85]
Backpage	http://backpage.com	[5]

should not expect the RNN to remember and consider the previous sentence's classification when it attempts to classify the next sentence. If the previous sentence's classification needs to be taken into account, the document should not be defined as a sentence but a paragraph, for instance. In that case, tokens could be defined as either terms or sentences. Despite the sequence of tokens, fed to the RNN, could be of arbitrary length, the RNN's memory is not limitless or perfect. For example, if our aim is to classify long documents and we define terms as tokens, we should not expect the RNN to fully remember how it classified the first token, when it reaches the last one. This problem is referred to as RNN's memory decay or vanishing gradient problem [10].

It is noteworthy that CNN is also capable of classifying a textual document as a whole [92], taking the contextual and structural information into account, but with one major limitation as opposed to RNN: all documents should contain the same number of tokens because the number of units at the CNN's input layer is fixed.. However, current implementations of RNN in libraries such as Keras on Python also require all documents to contain the same number of tokens. This requirement is usually addressed by extending smaller documents using blank words with zero-filled feature vectors. Each token forms one unit at the CNN's input layer. Elements in a token's feature vector are considered as different channels [92], the same way that colors are considered as different channels in image classification with CNN. The application of deep RNNs and CNNs has only been recently empowered by the growing size of training datasets and enhancements in computers' space and speed capacities. However, they are yet to be adequately applied to Web page classification.

4.5 Lack of a detailed testbed

Most studies use self-collected datasets to train and test their Web page classifiers. The lack of a comprehensive testbed makes it difficult to compare the accuracy of different Web page classifiers. Table 3 outlines studies that have applied publically available datasets to train and test their Web page classifiers. However, developing a detailed testbed along with evaluation metrics and establishing standard benchmarks remain a gap in assessing Web page classifiers.

5 Conclusions and future directions

In short, our study highlights the following limitations and possible future directions in:

a) classifying the textual content:

- metadata and contextual information surrounding the terms are mostly ignored in textual content classification,
- the structure and distribution of text in HTML tags and hyperlinks are understudied in textual content classification,

b) classifying the visual content:

- image classification methods rely heavily on computationally intensive and problem-specific analyses for feature extraction, e.g. detecting pornographic images rely on extracted features that are not suitable for other types of images

c) classifying the Web page as a whole:

- measuring the effectiveness of features in distinguishing among Web page classes or measuring the contribution of each feature in the classification accuracy is a prominent research gap,
- semi-supervised learning is understudied, despite its importance in Web page classification because of the tremendous amount of unlabeled Web pages and the high cost of labeling
- deep learning, CNN, RNN, and reinforcement learning remain underexplored but intriguing for Web page classification, and last but not least
- developing a detailed testbed along with evaluation metrics and establishing standard benchmarks remain a gap in assessing Web page classifiers

References

1. Abbasi, A., & Chen, H. (2007). Detecting fake escrow websites using rich fraud cues and kernel-based methods. *17th Annual Workshop on Information Technologies and Systems*, (pp. 55–60). Montreal, Canada.
2. Abbasi A, Chen H (2009) A comparison of tools for detecting fake websites. Computer 42(10):78–86
3. Abin, A. A., Fotouhi, M., & Kasaei, S. (2008). Skin segmentation based on cellular learning automata. *6th International Conference on Advances in Mobile Computing and Multimedia* (pp. 254-259). Linz, Austria: ACM.
4. Ahmadi A, Fotouhi M, Khaleghi M (2011) Intelligent classification of web pages using contextual and visual features. Appl Soft Comput 11(2):1638–1647
5. Alvari H, Shakarian P, Snyder JK (2017) Semi-supervised learning for detecting human trafficking. Security Informatics 6(1). https://doi.org/10.1186/s13388-017-0029-8
6. Ap-Apid, R. (2005). An algorithm for nudity detection. *5th Philippine Computing Science Congress*, (pp. 201-205).
7. Arentz WA, Olstad B (2004) Classifying offensive sites based on image content. Comput Vis Image Underst 94(1–3):295–310
8. Baecchi C, Uricchio T, Bertini M, Bimbo AD (2016) A multimodal feature learning approach for sentiment analysis of social network multimedia. Multimed Tools Appl 75(5):2507–2525
9. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural emachine translation by jointly learning to align and translate. *arXiv preprint* , arXiv:1409.0473.
10. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
11. Bosson, A., Cawley, G. C., Chan, Y., & Harvey, R. (2002). Non-retrieval: blocking pornographic images. *International Conference on Image and Video Retrieval* (pp. 50-60). Berlin, Heidelberg: Springer.

12. Chan, Y., Harvey, R., & Bangham, J. A. (2000). Using colour features to block dubious images. *10th European Signal Processing Conference. 3*, pp. 1-4. IEEE.
13. Chiu, J. P., & Nichols, E. (2015). Named entity recognition with bidirectional LSTM-CNNs. *arXiv preprint* , arXiv:1511.08308.
14. Chou N, Ledesma R, Teraguchi Y, Mitchell JC (2004) Client-side defense against web-based identity theft. In: 11th annual network and distributed system security symposium. Internet Society, San Diego
15. Chua CE, Wareham J (2004) Fighting internet auction fraud: an assessment and proposal. Computer 37(10):31–37
16. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12(Aug):2493–2537
17. Denoyer L, Gallinari P (2004) Bayesian network model for semi-structured document classification. Inf Process Manag 40(5):807–827
18. Diligenti, M., Gori, M., Maggini, M., & Scarselli, F. (2001). Classification of html documents by hidden tree-markov models. *Sixth International Conference on Document Analysis and Recognition* (pp. 849-853). Seattle, WA, USA: IEEE.
19. Du, R., Safavi-Naini, R., & Susilo, W. (2003). Web filtering using text classification. *The 11th IEEE International Conference on Networks* (pp. 325-330). IEEE.
20. Dumais, S., & Chen, H. (2000). Hierarchical classification of Web content. *23rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 256-263). ACM.
21. Fakeri-Tabrizi A, Amini M-R, Goutte C, Usunier N (2015) Multiview self-learning. Neurocomputing 155(1):117–127
22. Farfade, S. S., Saberian, M. J., & Li, L.-J. (2015). Multi-view face detection using deep convolutional neural networks. *5th International Conference on Multimedia Retrieval* (pp. 643-650). ACM.
23. Fauzi F, Belkhatir M (2010) A user study to investigate semantically relevant contextual information of WWW images. International Journal of Human-Computer Studies 68(5):270–287
24. Fauzi F, Belkhatir M (2013) Multifaceted conceptual image indexing on the world wide web. Inf Process Manag 49(2):420–440
25. Fersini E, Messina E, Archetti F (2008) Enhancing web page classification through image-block importance analysis. Inf Process Manag 44(4):1431–1447
26. Forsyth DA, Fleck MM (1999) Automatic detection of human nudes. Int J Comput Vis 32(1):63–77
27. Hammami, M., Chahir, Y., & Chen, L. (2003). WebGuard: web based adult content detection and filtering system. *IEEE/WIC International Conference on Web Intelligence* (pp. 574-578). IEEE.
28. Hammami M, Chahir Y, Chen L (2006) Webguard: a web filtering engine combining textual, structural, and visual content-based analysis. IEEE Trans Knowl Data Eng 18(2):272–284
29. Hashemi M, Hall M (2018) Visualization, feature selection, machine learning: identifying the responsible group for extreme acts of violence. IEEE Access 6(1):70164–70171
30. Hashemi, M., & Hall, M. (2018). Identifying the responsible group for extreme acts of violence through pattern recognition. *International Conference on HCI in Business, Government, and Organizations* (pp. 594-605). Cham: Springer.
31. Hashemi M, Hall M (2019) Detecting and classifying online dark visual propaganda. Image Vis Comput 89:95–105
32. Ho, W. H., & Watters, P. A. (2004). Statistical and structural approaches to filtering internet pornography. *IEEE International Conference on Systems, Man and Cybernetics. 5*, pp. 4792-4798. IEEE.
33. Howard, A. G. (2013). Some improvements on deep convolutional neural network based image classification. *arXiv preprint* , arXiv:1312.5402.
34. Hu W, Wu O, Chen Z, Fu Z, Maybank S (2007) Recognition of pornographic web pages by classifying texts and images. IEEE Trans Pattern Anal Mach Intell 29(6):1019–1034
35. Hu W, Zuo H, Wu O, Chen Y, Zhang Z, Suter D (2011) Recognition of adult images, videos, and web page bags. ACM Transactions on Multimedia Computing, Communications, and Applications 7(1):28
36. Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint* , arXiv:1508.01991.
37. Ioffe, S., & Forsyth, D. (1999a). Finding people by sampling. *The Seventh IEEE International Conference on Computer Vision. 2*, pp. 1092-1097. IEEE.
38. Ioffe, S., & Forsyth, D. A. (1999b). Learning to find pictures of people. *Advances in Neural Information Processing Systems. 11*, pp. 782-788. MIT Press.
39. Ioffe S, Forsyth DA (2001) Probabilistic methods for finding people. Int J Comput Vis 43(1):45–68
40. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. *The 22nd ACM International Conference on Multimedia* (pp. 675-678). ACM.

41. Jiao, F., Gao, W., Duan, L., & Cui, G. (2001). Detecting adult image using multiple features. *International Conference on Info-tech and Info-net Proceedings. 3*, pp. 378-383. Beijing: IEEE.
42. JingHua B, Xian ZX, ZhiXin L, XiaoPing L (2012) Mixture models for web page classification. Phys Procedia 25(1):499–505
43. Jones MJ, Rehg JM (2002) Statistical color models with application to skin detection. Int J Comput Vis 46(1):81–96
44. Jurafsky D, Martin JH (2014) Speech and language processing. Pearson, London
45. Kim S, Zhang B-T (2003) Genetic mining of HTML structures for effective web-document retrieval. Appl Intell 18(3):243–256
46. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, (pp. 1097-1105).
47. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint* , arXiv:1603.01360.
48. Lee, J. Y., & Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint* , arXiv:1603.03827.
49. Lee PY, Hui SC, Fong AC (2002) Neural networks for web content filtering. IEEE Intell Syst 17(5):48–57
50. Lee PY, Hui SC, Fong AC (2005) An intelligent categorization engine for bilingual web content filtering. IEEE Transactions on Multimedia 7(6):1183–1190
51. Lee J-H, Yeh W-C, Chuang M-C (2015) Web page classification based on a simplified swarm optimization. Appl Math Comput 270(1):13–24
52. Li L, Helenius M (2007) Usability evaluation of anti-phishing toolbars. J Comput Virol 3(2):163–184
53. Li H, Xu Z, Li T, Sun G, Choo K-KR (2017a) An optimized approach for massive web page classification using entity similarity based on semantic network. Futur Gener Comput Syst 76(1):510–518
54. Li X, Rao Y, Xie H, Lau RY, Yin J, Wang FL (2017b) Bootstrapping social emotion classification with semantically rich hybrid neural networks. IEEE Trans Affect Comput 8(4):428–442
55. Liang, K. M., Scott, S. D., & Waqas, M. (2004). Detecting pornographic images. *Asian Conference on Computer Vision*, (pp. 497-502).
56. Liparas, D., HaCohen-Kerner, Y., Moumtzidou, A., Vrochidis, S., & Kompatsiaris, I. (2014). News articles classification using random forests and weighted multimodal features. *Information Retrieval Facility Conference* (pp. 63-75). Springer.
57. Liu W, Deng X, Huang G, Fu AY (2006) An antiphishing strategy based on visual similarity assessment. IEEE Internet Comput 10(2):58–65
58. Luo Y (2017) Recurrent neural networks for classifying relations in clinical notes. J Biomed Inform 72(1): 85–95
59. McKenna SJ, Gong S, Raja Y (1998) Modelling facial colour and identity with gaussian mixtures. Pattern Recogn 31(12):1883–1892
60. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *11th Annual Conference of the International Speech Communication Association*, *2*, p. 3.
61. Moustafa, M. (2015). Applying deep learning to classify pornographic images and videos. *7th Pacific-Rim Symposium on Image and Video Technology* (p. arXiv:1511.08899). At Auckland, New Zealand: arXiv preprint.
62. Munkhdalai, T., & Yu, H. (2016a). Reasoning with memory augmented neural networks for language comprehension. *arXiv preprint* , arXiv:1610.06454.
63. Munkhdalai, T., & Yu, H. (2017a). Neural semantic encoders. *The Annual Meeting of the Association for Computational Linguistics. 1*, pp. 397-407. HHS Public Access.
64. Munkhdalai, T., & Yu, H. (2017b). Neural tree indexers for text understanding. *The Annual Meeting of the Association for Computational Linguistics. 1*, pp. 11-21. HHS Public Access.
65. Munkhdalai, T., Lalor, J., & Yu, H. (2016b). Citation analysis with neural attention models. *The 7th International Workshop on Health Text Mining and Information Analysis*, (pp. 69-77).
66. Nian F, Li T, Wang Y, Xu M, Wu J (2016) Pornographic image detection utilizing deep convolutional neural networks. Neurocomputing 210(1):283–293
67. Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1717-1724). IEEE.
68. Özel SA (2011) A web page classification system based on a genetic algorithm using tagged-terms as features. Expert Syst Appl 38(4):3407–3415
69. Perez M, Avila S, Moreira D, Moraes D, Testoni V, Valle E et al (2017) Video pornography detection through deep learning techniques and motion information. Neurocomputing 230(1):279–293
70. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137

71. Rajaraman, A., & Ullman, J. D. (2011). Data mining. In *Mining of Massive Datasets* (pp. 1-17). Cambridge University Press.
72. Ribeiro, A., Fresno, V., Garcia-Alegre, M. C., & Guinea, D. (2003). Web page classification: a soft computing approach. *International Atlantic Web Intelligence Conference* (pp. 103-112). Berlin, Heidelberg: Springer.
73. Rowley HA, Jing Y, Baluja S (2006) Large scale image-based adult-content filtering. In: International conference on computer vision theory and applications, 1, pp 290–296
74. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Process Manag 24(5):513–523
75. Segalin C, Cheng DS, Cristani M (2017b) Social profiling through image understanding: personality inference using convolutional neural networks. Comput Vis Image Underst 156(1):34–50
76. Selamat A, Omatu S (2004) Web page feature selection and classification using neural networks. Inf Sci 158(1):69–88
77. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint* , arXiv:1409.1556.
78. Sun W, Su F, Wang L (2018) Improving deep neural networks with multi-layer maxout networks and a novel initialization method. Neurocomputing 278(1):34–40
79. Sundermeyer, M., Alkhouli, T., Wuebker, J., & Ney, H. (2014). Translation modeling with bidirectional recurrent neural networks. *The Conference on Empirical Methods in Natural Language Processing*, (pp. 14-25).
80. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9). IEEE.
81. Tamura A, Watanabe T, Sumita E (2014) Recurrent neural networks for word alignment model. *The 52nd Annual Meeting of the Association for*. Computational Linguistics 1:1470–1480
82. Terrillon, J. C., Shirazi, M. N., Fukamachi, H., & Akamatsu, S. (2000). Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. *Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 54-61). IEEE.
83. Tian L, Zheng D, Zhu C (2013) Image classification based on the combination of text features and visual features. Int J Intell Syst 28(3):242–256
84. Trotman A (2005) Choosing document structure weights. Inf Process Manag 41(2):243–264
85. Ulges, A., & Stahl, A. (2011). Automatic detection of child pornography using color visual words. *IEEE International Conference on Multimedia and Expo* (pp. 1-6). IEEE.
86. Uysal AK, Gunal S (2014) The impact of preprocessing on text classification. Inf Process Manag 50(1): 104–112
87. Wang D, Nyberg E (2015) A long short-term memory model for answer sentence selection in question answering. *The 53rd Annual Meeting of the Association for*. Computational Linguistics 2:707–712
88. Wang, J. Z., Wiederhold, G., & Firschein, O. (1997). System for screening objectionable images using daubechies' wavelets and color histograms. *International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services* (pp. 20-30). Berlin, Heidelberg: Springer.
89. Wang JZ, Li J, Wiederhold G, Firschein O (1998) System for screening objectionable images. Comput Commun 21(15):1355–1360
90. Wang M, Liu X, Wu X (2015) Visual classification by l1-hypergraph modeling. IEEE Trans Knowl Data Eng 27(9):2564–2574
91. Wang, X., Cheng, F., Wang, S., Sun, H., Liu, G., & Zhou, C. (2018). Adult image classification by a local-context aware network. *25th IEEE International Conference on Image Processing* (pp. 2989-2993). Athens, Greece: IEEE.
92. Xiong S, Lv H, Zhao W, Ji D (2018) Owards twitter sentiment classification by multi-level sentiment-enriched word embeddings. Neurocomputing 275(1):2459–2466
93. Xu Y, Li B, Xue X, Lu H (2005) Region-based pornographic image detection. *IEEE 7th Workshop on Multimedia Signal Processing* (pp. 1–4). IEEE, Shanghai, China
94. Yan, X., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency path. *arXiv preprint* , arXiv:1508.03720.
95. Yang Y, Slattery S, Ghani R (2002) A study of approaches to hypertext categorization. J Intell Inf Syst 18(2–3):219–241
96. Yang X, Zhang T, Xu C (2015) Cross-domain feature learning in multimedia. IEEE Transactions on Multimedia 17(1):64–78
97. Yi, J., & Sundaresan, N. (2000). A classifier for semi-structured documents. *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 340-344). ACM.

98. Yu J, Tao D, Wang M (2012) Adaptive hypergraph learning and its application in image classification. IEEE Trans Image Process 21(7):3262–3272
99. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision* (pp. 818-833). Springer.
100. Zhang Y, Egelman S, Cranor L, Hong J (2007) Phinding phish: evaluating anti-phishing tools. In: 14th Annual Network & Distributed System Security Symposium. Internet Society, San Diego, CA
101. Zhang H, Liu G, Chow TW, Liu W (2011) Textual and visual content-based anti-phishing: a Bayesian approach. IEEE Trans Neural Netw 22(10):1532–1546
102. Zhao XG, Wang G, Bi X, Gong P, Zhao Y (2011) XML document classification based on ELM. Neurocomputing 74(16):2444–2451
103. Zheng, H., Liu, H., & Daoudi, M. (2004). Blocking objectionable images: adult images and harmful symbols. *IEEE International Conference on Multimedia and Expo. 2*, pp. 1223-1226. IEEE.
104. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., et al. (2016). Attention-based bidirectional long short-term memory networks for relation classification. *The 54th Annual Meeting of the Association for Computational Linguistics*, *2*, pp. 207-212.
105. Mahdi Hashemi, (2019) Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. Journal of Big Data 6 (1)
106. Hashemi, M., Hall, M. (2020). Criminal tendency detection from facial images and the gender bias effect. Journal of Big Data 7 (2)

Mahdi Hashemi , Ph.D. is an Assistant Professor in the Department of Information Sciences and Technology at George Mason University. He leads the Machine Learning and Smart Cities Group, where he also specializes in intelligent transportation, spatial-temporal data and web/social media analytics.

Affiliations

Mahdi Hashemi [1]

[1] Department of Information Sciences and Technology, George Mason University, 4400 University Dr, Fairfax, VA 22030, USA