Web-page Classification through Summarization

Dou Shen[1] Zheng Chen[2] Qiang Yang[3] Hua-Jun Zeng[2] Benyu Zhang[2] Yuchang Lu[1] Wei-Ying Ma[2]

[1]Computer Science and Tech.
Tsinghua University
Beijing, P.R.China
shendou@mails.tsinghua.edu.cn,
lyc@mail.tsinghua.edu.cn

[2]Microsoft Research Asia
49 Zhichun Road
Beijing, P.R.China
{zhengc, hjzeng, byzhang,
wyma}@microsoft.com

[3] Hong Kong University of
Science and Technology
Clearwater Bay
Kowloon, Hong Kong
qyang@cs.ust.hk

ABSTRACT

Web-page classification is much more difficult than pure-text classification due to a large variety of noisy information embedded in Web pages. In this paper, we propose a new Web-page classification algorithm based on Web summarization for improving the accuracy. We first give empirical evidence that ideal Web-page summaries generated by human editors can indeed improve the performance of Web-page classification algorithms. We then propose a new Web summarization-based classification algorithm and evaluate it along with several other state-of-the-art text summarization algorithms on the LookSmart Web directory. Experimental results show that our proposed summarization-based classification algorithm achieves an approximately 8.8% improvement as compared to pure-text-based classification algorithm. We further introduce an ensemble classifier using the improved summarization algorithm and show that it achieves about 12.9% improvement over pure-text based methods.

Categories and Subject Descriptors

H.4.m [**Information Systems Application**]: Miscellaneous; I.5.4 [**Pattern Recognition**]: Applications-*Text processing*;

General Terms

Algorithms, Experimentation, Verification.

Keywords

Web Page Categorization, Web Page Summarization, Content Body

1. INTRODUCTION

With the rapid growth of the World Wide Web (WWW), there is an increasing need to provide automated assistance to Web users for Web page classification and categorization. Such an assistance is helpful in organizing the vast amount of information returned by keyword-based search engines, or in constructing catalogues that organize Web documents into hierarchical collections; examples of the latter include the Yahoo (http://www.yahoo.com) directory and the LookSmart directory (http://search.looksmart.com). There is evidence that categorization is expected to play an important role in future search services. For example, research conducted by Chen and Dumais shows that users prefer navigating through catalogues of pre-classified content [6]. Such a strong need, however, is difficult to meet without automated Web-page classification techniques due to the labor-intensive nature of human editing.

On a first glance, Web-page classification can borrow directly from the machine learning literature for text classification [21][24][27]. On closer examination, however, the solution is far from being so straightforward. Web pages have their own underlying embedded structure in the HTML language. They typically contain noisy content such as advertisement banner and navigation bar. If a pure-text classification method is directly applied to these pages, it will incur much bias for the classification algorithm, making it possible to lose focus on the main topics and important content. Thus, a critical issue is to design an intelligent preprocessing technique to extract the main topic of a Web page.

In this paper, we show that using Web-page summarization techniques for preprocessing in Web-page classification is a viable and effective technique. We further show that instead of using an off-the-shelf summarization technique that is designed for pure-text summarization, it is possible to design specialized summarization methods catering to Web-page structures. In order to collect the empirical evidence that summarization techniques can benefit Web classification, we first conduct an ideal case experiment, in which each Web page is substituted by its summary generated by human editors. Compared to using the full-text of the Web pages, we gain an impressive 14.8% improvement in F1 measurement. In addition, in this paper, we also propose a new automatic Web summarization algorithm, which extracts the main topic of a Web page by a page-layout analysis to enhance the accuracy of classification. We evaluate the classification performance with this algorithm and compare to some traditional state-of-the-art automatic text summarization algorithms including supervised methods and unsupervised learning methods. Experiment results on LookSmart Web directory show that all summarization methods can improve the micro F1 measure. Finally, we show that an ensemble of summarization methods can achieve about 12.9% improvement relatively on micro F1 measure, which is very close to the upper bound achieved in our ideal case experiment.

The rest of the paper is organized as follows. In Section 2, we present the related works on Web classification and summarization. Then we present our proposed unsupervised and supervised summarization algorithms in Section 3. In Section 4, the experimental results on LookSmart Web directory are shown

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

as well as some discussions. Finally, we conclude our work in Section 5.

2. RELATED WORK

Recently much work has been done on Web-page summarization [2][4][12]. Ocelot [2] is a system for summarizing Web pages using probabilistic models to generate the "gist" of a Web page. The models used are automatically obtained from a collection of human-summarized Web pages. In [4], Buyukkokten et al. introduces five methods for summarizing parts of Web pages on handheld devices where the core algorithm is to compute the words' importance using TF/IDF measures and to select important sentences using Luhn's classical method [23]. In [12], Delort exploits the effect of context in Web page summarization, which consists of the information extracted from the content of all the documents linking to a page. It is shown that summaries that take into account of the context information are usually more relevant than those made only from the target document.

Some research has been done to enhance categorization by summarization [17][18][19], but these works handle pure text categorization only. In [19], Kolcz et al. uses summarization as a feature selection method and applies a simple extraction-based technique with several heuristic rules.

Our work is related to that for removing noise from a Web page. In this aspect, Yi et al. propose an algorithm by introducing a tree structure, called Style Tree, to capture the common presentation styles and the actual contents of the pages in a given Web site [33]. However, the Style Tree is difficult to be built when the number of Web sites is large.

The structure of a Web page is influenced by many factors. Chen et al. pointed out in [8] that when authoring a Web site, the editors usually first conceive the information structure of the site in their mind. They then compile their thoughts into cross-linked Web pages by HTML language and finally, some extra information, such as navigation bar, advertisement, and copyright information are inserted to prettify the whole page. Since HTML is a visual representation language, much useful information about the content organization is lost after the authoring step. In order to find the important structural information again, two methods have been widely used. One is to extract title and meta-data included in HTML tags to represent the semantic meaning of the Web pages. It's usually true that title and meta-data should be good information to be used by their authors to indicate the main content of Web pages. However we can not fully rely on them due to the following reasons. First, title and meta-data may be empty in some Web pages. For example, in our dataset, about 24.6% of the pages are without any meta-data and 4.8% pages are without a title. Second, some of titles and meta-data may be meaningless since Website designer may not fill them in and may simply set them by default, with such useless names as "page1", "page2". Finally, Web site designers may misuse or even give the wrong title or meta-data fields to cheat search engines in order to boost up their ranking.

Therefore, it is critical for us to extract the main topic of a Web page by automatically analyzing their context features, such as the anchor text pointing to a Web page [1][5][13]. In this direction, Glover et al. [13] provided an analysis of the utility of text in citing documents for classification and proved that anchor text was valuable. Nevertheless, this should be done with care; Chakrabarti [5] studied the role of hyperlink in hypertext classification and pointed out that a naïve use of terms in the linked neighborhood of a Web page could even *degrade* the classification performance.

To summarize, our aim is to apply Web-page summarization to Web-page classification, rather than using pure-text summarization for the purpose. We will show that the special nature of Web pages have a large impact on the classification performance.

3. WEB-PAGE SUMMARIZATION

In this section, we consider how to analyze the complex implicit structure embedded in Web pages, and how to use this information for summarization of Web pages. Our approach is to extract most relevant contents from the Web pages and then pass them on to a standard text classification algorithm.

In particular, we will consider four different methods for conducting the Web page summarization. The first method corresponds to an adaptation of Luhn's summarization technique. The second method corresponds to using Latent Semantic Analysis on Web pages for summarization. The third method corresponds to finding the important content body as a basic summarization component. Finally, the fourth method looks at summarization as a supervised learning task. We combine the results of all four summarization methods into an ensemble of summarizers, and use it for Web page summarization.

3.1 Adapted Luhn's Summarization Method

We adapt Luhn's method that is designed text summarization for the purpose of Web-page summarization. Luhn's method is a systematic approach to perform summarization which forms the core of the field today [23]. In this extraction-based method, every sentence is assigned with a significance factor, and the sentences with the highest significance factor are selected to form the summary. In order to compute the significance factor of a sentence, we need to build a "significant words pool" which is defined as those words whose frequency is between high-frequency cutoff and low-frequency cutoff that can be tuned to alter the characteristics of the summarization system. After this is done, the significant factor of a sentence can be computed by Luhn's method as follows: (1) set a limit L for the distance at which any two significant words could be considered as being significantly related. (2) find out a portion in the sentence that is bracketed by significant words not more than L non-significant words apart. (3) count the number of significant words contained in the portion and divide the square of this number by the total number of words within the portion. The result is the significant factor related to S.

In order to customize this procedure for Web-pages, we make a modification to Luhn's algorithm. In our Web classification task, the category information of each page is already known in the training data, thus significant-words selection could be processed within each category. In this way, we build significant words pool for each category by selecting the words with high frequency after removing the stop words in that category and then employing Luhn's method to compute the significant factor.

There are two advantages for this modification. First, the prior knowledge of categories is utilized in summarization. Second, some noisy words which may be relatively frequent in an individual page will be removed through the use of statistics over multiple documents. When summarizing the Web pages in the training set, the significant score of each sentence is calculated according to the significant-words pool corresponding to its category label. For a testing Web page, we do not have the category information. In this case, we will calculate the significant factor for each sentence according to different significant words pools over *all* categories separately. The significant score for the target sentence will be averaged over all categories and referred to as S_{luhn}. The summary of this page will be formed by the sentences with the highest scores.

3.2 Latent Semantic Analysis (LSA)

Latent Semantics Analysis (LSA) has been successfully applied to information retrieval [11] as well as many other related domains. Its power is derived from its ability to represent terms and related concepts as points in a very high dimensional "semantic space" [22]. In the text summarization area, Gong [14] is one of the works that has successfully applied the LSA to pure text. In this section, we will review how to apply LSA to summarization.

To begin with, LSA is based on singular value decomposition (SVD), a mathematical matrix decomposition technique that is applicable to text corpora experienced by people. Given an *m*n* matrix *A = [A1,A2,...An]*, with each column vector *Ai* representing the weighted term-frequency vector of sentence *i* in the document under consideration, the SVD is defined as:

$$A = U\Sigma V^T$$

where $U = [u_{ij}]$ is an *m*n* column-orthonormal matrix whose columns are called *left singular vectors*; $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an n*n diagonal matrix whose diagonal elements are non-negative singular values sorted in descending order. $V = [v_{ij}]$ is an $n \times n$ othonormal matrix whose columns are called right singular vectors.[26]

As noted in [3][11], LSA is applicable in summarization because of two reasons. First, LSA is capable of capturing and modeling interrelationships among terms by semantically clustering terms and sentences. Second, LSA can capture the salient and recurring word combination pattern in a document which describes a certain topic or concept. In LSA, concepts are represented by one of the singular vectors where the magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentence containing this word combination pattern will be projected along this singular vector. The sentence that best represents this pattern will have the largest index value with this vector. We denote this index value as S_{lsa} and select the sentences with the highest S_{lsa} to form the summary. The pseudo-code of SVD-based summarization method can be found in [14].

3.3 Content Body Identification by Page Layout Analysis

The structured character of Web pages makes Web-page summarization different from pure-text summarization. This task is difficult due to a number of "noisy" components on a Web page, such as the navigation bar, advertisement, and copyright information. In order to utilize the structure information of Web pages, we employ a simplified version of the Function-Based Object Model (FOM) as described in [7].

In a nutshell, FOM attempts to understand an authors' intention by identifying the object's function and category. In FOM, objects are classified into a Basic Object (BO), which is the smallest information body that cannot be further divided, or a Composite Object (CO) which is a set of Objects (BO or CO) that perform some functions together. An example of a BO is a jpeg file. In HTML contents, a BO is a non-breakable element within two tags or an embedded object. There is no other tag inside the content of a BO. According to this criterion, it is easy to find out all the BOs inside a Web page. Likewise, COs can be detected by a layout analysis of Web pages. The basic idea is that objects in the same category generally have consistent visual styles so that they are separated by apparent visual boundaries, such as table boundaries, from the objects in other categories.

After detecting all the BOs and COs in a Web page, we could identify the category of each object according to some heuristic rules. Detailed examples of these rules are shown in [7]; here we give an overview only. First, the categories of objects include:

1) Information Object: this object presents content information.

2) Navigation Object: this object provides navigation guide.

3) Interaction Object: this object provides user side interaction.

4) Decoration Object: this object serves for decoration purpose.

5) Special Function Object: this object performs special functions such as AD, Logo, Contact, Copyright, Reference, etc.

In order to make use of these objects, from the above types of objects, we define the *Content Body* (CB) of a Web page which consists of the main objects related to the topic of that page; these are the objects that convey important information about the page. The algorithm for detecting CB is as follows:

1. Consider each selected object as a single document and build the TF*IDF index for the object.
2. Calculate the similarity between any two objects using Cosine similarity computation, and add a *link* between them if their similarity is greater than a threshold. The threshold is chosen empirically. After processing all pairs of objects, we will obtain a linked graph to connect different objects.
3. In the graph, a *core object* is defined as the object having the most edges.
4. Extract the *CB* as the combination of all objects that have edges linked to the *core object.*

Finally, we will assign a score S_{cb} to each sentence, for which $S_{cb} = 1.0$ if the sentence is included in "content body"; otherwise, $S_{cb} = 0.0$. Finally, all sentences with S_{cb} equal to 1.0 give rise to the summary of the Web page in question.

3.4 Supervised Summarization

Besides the unsupervised summarization algorithms described above, some researchers also focus on generating the summary using machine learning approaches [2][9][20][30]. In this paper,

we also employ a supervised approach for Web summarization, by making full use of the labeled training data. A set of features are first extracted from each of a Web page. Then, a supervised learning algorithm is applied to train the summarizer to identify whether a sentence should be selected into its summary or not. There are a total of eight features utilized in our algorithm, where five of them are common features for text document and Web page and the rest three of them are specific to Web page layout.

Some notations are defined as follows:

PN: the number of paragraphs in a document;

SN: the number of sentences in a document;

PL_k: the number of sentences in a certain paragraph k

Para(i): the associated paragraph of sentence i

TF_w: the number of occurrences of word w in a target Web page;

SF_w: the number of sentences including the word w in the b page;

Given a set of sentences S_i (i = 1… SN) in a page, the eight features are defined as follows:

(1) f_{i1} measures the position of a sentence S_i in a certain paragraph.

(2) f_{i2} measures the length of a sentence S_i, which is the number of words in S_i.

(3) $f_{i3}= \sum TF_w * SF_w$. This feature takes into account not only the number of word w into consideration, but also its distribution among sentences. We use it to punish the locally frequent words.

(4) f_{i4} is the similarity between S_i and the title. This similarity is calculated as the dot product between the sentence and the title.

(5) f_{i5} is the cosine similarity between S_i and all text in the page.

(6) f_{i6} is the cosine similarity between S_i and meta-data in the page.

(7) f_{i7} is the number of occurrences of word from S_i in special word set. The special word set is built by collecting the words in the Web page that are italic or bold or underlined.

(8) f_{i8} is the average font size of the words in S_i. In general, larger font size in a Web page is given higher importance.

After extracting these eight features from a Web page, we apply the Naïve Bayesian classifier to train a summarizer, as in [20].

$$p(s \in S \mid f_1, f_2 \ldots f_8) = \frac{\prod_{j=1}^{8} p(f_j \mid s \in S) p(s \in S)}{\prod_{j=1}^{8} p(f_j)}$$

where $p(s \in S)$ stands for the compression rate of the summarizer, which can be predefined for different applications, $p(f_i)$ is the probability of each feature i and $p(f_i \mid s \in S)$ is the conditional probability of each feature i. The latter two factors can be estimated from the training corpus. Each sentence will then be assigned a score by the above equation, which is denoted as $S_{\sup}$.

3.5 An Ensemble of Summarizers

By combining the four methods presented in the previous sections, we obtain a hybrid Web-page. Given an incoming Web page, we calculate the importance score for each sentence by the four summarization algorithms separately. The final score of a sentence is the sum of the four scores.

$$S = S_{luhn} + S_{lsa} + S_{cb} + S_{\sup}$$

The sentences with the highest S will be chosen into the summary.

4. EXPERIMENTS

In order to test the effectiveness of summarization for Web classification, several experiments are conducted. Firstly, we test the Web page classification on the *human created summaries* in order to find out whether the summarization can help classification of Web pages at all. Having confirmed this hypothesis, we compare our proposed "content body identification summarizer" with two traditional algorithms: adapted Luhn's algorithm and LSA-based methods, as well as the supervised summarizers. Finally, our ensemble of summarizers is evaluated. In our experiments, we also study the variation of different parameter settings for composing the best summarizer.

4.1 Data Set

In our experiments, we use about 2 millions Web pages crawled from the LookSmart Web directory (http://search.looksmart.com). Due to the limitation of network bandwidth, we only downloaded about 500 thousand descriptions of Web pages that are manually created by human editors. Since it is a time-consuming task to run experiments on this large data set, we randomly sampled 30% of the pages with descriptions for our experiment purpose. The extracted subset includes 153,019 pages, which are distributed among 64 categories (we only consider the top two level categories on LookSmart Website). The largest category (Library\Society) consists of 17,473 pages; while the smallest category (People & Chat\Find People) consists of only 52 pages. Table 1 and Table 2 show the number of pages for the three largest categories and three smallest categories. In order to reduce the uncertainty of data split, a 10-fold cross validation procedure is applied in our experiments.

Table 1. The Three largest categories

Category Name	Total	Train	Test
Library\Society	17473	15726	1747
Travel\Destinations	13324	11992	1332
Entertainment\Celebrities	10112	9101	1011

Table 2. The Three smallest categories

Category Name	Total	Train	Test
Sports\News & Scores	106	96	10
People & Chat\Personals	74	67	7
People & Chat\Find People	52	47	5

4.2 Classifiers

Since the focus of this paper is to test the effectiveness of Web summarization for classification, we choose two popular classifiers in our experiments. One is a naïve Bayesian classifier [24] [25], and another is a support vector machine [10][15][32].

4.2.1 Naïve Bayesian Classifier (NB)

The Naïve Bayesian Classifier (NB) is a simple but effective text classification algorithm which has been shown to perform very well in practice [24] [25]. The basic idea in NB is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. As described in [24], most researchers employ NB method by applying Bayes' rule:

$$P(c_j \mid d_i;\hat{\theta}) = \frac{P(c_j \mid \hat{\theta})\prod_{k=1}^{n} P(w_k \mid c_j;\hat{\theta})^{N(w_k,d_i)}}{\sum_{r=1}^{|C|} P(c_r \mid \hat{\theta})\prod_{k=1}^{n} P(w_k \mid c_r;\hat{\theta})^{N(w_k,d_i)}}$$

where $P(c_j \mid \hat{\theta})$ can be calculated by counting the frequency with each category c_j occurring in the training data; $|C|$ is the number of categories; $p(w_i \mid c_j)$ stands for probability that word w_i occurs in class c_j which maybe small in training data, so the Laplace smoothing is chosen to estimate it; $N(w_k, d_i)$ is the number of occurrences of a word w_k in d_i; n is the number of words in the training data.

4.2.2 Support Vector Machine (SVM)

Support vector machine (SVM) is a powerful learning method recently introduced by V.Vapnik et al. [10][15][32]. It is well founded in terms of computational learning theory and has been successfully applied to text categorization [15] [16].

SVM operates by finding a hyper-surface in the space of possible inputs. The hyper-surface attempts to split the positive examples from the negative examples by maximizing the distance between the nearest of the positive and negative examples to the hyper-surface. Intuitively, this makes the classification correct for testing data that is near but not identical to the training data. There are various ways to train SVMs. One particularly simple and fast method is Sequential Minimal Optimization (SMO) developed by J. Platt which is available on [28]. His sequential minimal optimization algorithm breaks the large quadratic programming (QP) problem down into a series of small QP problems to be solved analytically. Thus the SMO algorithm is efficiently applicable for large feature and training sets.

4.3 Evaluation Measure

We employ the standard measures to evaluate the performance of Web classification, i.e. precision, recall and F1-measure [31]. Precision (*P*) is the proportion of actual positive class members returned by the system among all predicted positive class members returned by the system. Recall (R) is the proportion of predicted positive members among all actual positive class members in the data. F1 is the harmonic average of precision and recall as shown below:

$$F1 = 2 \times P \times R / (P + R)$$

To evaluate the average performance across multiple categories, there are two conventional methods: micro-average and macro-average. Micro-average gives equal weight to every document; while macro-average gives equal weight to every category, regardless of its frequency. In our experiments, only micro-average will be used to evaluate the performance of classification.

4.4 Experimental Results and Analysis

4.4.1 Baseline

A simple way to perform Web classification is to treat it as a pure-text document. In our experiment, the state-of-the-art text classification algorithms (NB & SVM) are applied to build the baseline system. Firstly, Web pages are converted to pure text document by removing the HTML tags. Then, each document is tokenized with a stop-word remover and Porter stemming [29]. Finally, each Web page is represented as a bag-of-words, in which the weight of each word is assigned with their term frequency [1]. In order to speed-up the classification, a simple feature selection method, "document frequency selection (DF)" [34], is applied in our experiment. In our experiments, the words whose DF is lower than six are removed from feature set. Finally, we obtain the classification results based on the selected word features, as shown in the "Full-text" row of Table 3 and Table 4. From these two tables, we found that SVM achieves 0.651 in micro-F1, which outperform the NB's result by about 2.4% relatively. We also found that the variance of 10-fold cross validation is quite small (about 0.3%), which indicates that the classification is stable on this dataset.

Table 3. Experimental results on NB

	microP	microR	micro-F1
Full-text	70.7±0.3	57.7±0.3	63.6±0.3
Title	68.3±0.4	55.4±0.4	61.2±0.4
Meta-data	47.7±0.4	38.7±0.4	42.7±0.4
Description	**81.5±0.4**	**66.2±0.4**	**73.0±0.4**
Content Body	77.2±0.4	62.7±0.4	69.2±0.4
Luhn	77.9±0.4	63.3±0.4	69.8±0.5
LSA	75.9±0.4	61.7±0.4	68.1±0.5
Supervised	75.2±0.4	60.9±0.4	67.3±0.4
Hybrid	80.2±0.3	65.0±0.3	71.8±0.3

Table 4. Experimental results on SMOX

	microP	microR	micro-F1
Full-text	72.4±0.3	59.3±0.3	65.1±0.3
Title	68.8±0.3	55.9±0.3	61.7±0.3
Meta-data	47.8±0.4	38.8±0.4	42.8±0.4
Description	**82.1±0.4**	**66.9±0.4**	**73.7±0.4**
Content Body	78.6±0.3	63.7±0.3	70.3±0.3
Luhn	77.3±0.3	62.8±0.3	69.3±0.3
LSA	79.2±0.3	64.3±0.3	71.0±0.3
Supervised	76.3±0.4	61.8±0.4	68.3±0.4
Hybrid	81.1±0.3	65.7±0.3	72.6±0.3

[1] We do not use the *tf***idf* weighting schema for the bag-of-words model since *tf* is informative enough and it is time consuming to calculate the inverted document frequency for large datasets.

4.4.2 Results on human's summary

In order to test the effectiveness of summarization techniques for Web classification, we conduct a feasibility study in our experiment. We extract the description of each Web page from the LookSmart Website and consider it as the "ideal" summary for the page. Since the description is authored by the Web directory editors, the quality is considered to be good enough to be the summary for the page. We apply the classifiers directly on these descriptions instead of the full text of the Web pages. This experiment can help us understand whether in the best case, summarization can help improve the classification. In addition, the title and meta-data of a Web page can also be considered as a kind of summary. An example of the description, title and meta-data is shown in Figure 1 and the classification results on these "ideal summary" are shown in the related rows of Table 3 and Table 4. Compared to full-text classification, classification on human-authored "description" can significantly improve the F1 measure by more than 13.2% using either classifier. However, classification on "pure title" or "pure meta-data" achieves worse F1-measure results as compared to the baseline system; this is because these descriptions are usually short and do not contain sufficient information. Through analyzing on the special cases, we found that Web-page "descriptions" can easily help the end-user to understand the meaning of the Web page. Although the title can play this role also to some extend, their short lengths is indeed impossible to represent the full meaning of the page. The uneven quality of the meta-data because some of them are the default values, also prevents them from achieving good results.

Through the "ideal case" experiments, we have found that the "ideal summary" can indeed help improve the Web classification performance. In addition, if the summary is not done properly, then the "bad summary" can hurt the performance. Hence, in the rest of the experiments, we hope to achieve a similar "good" summary by our automatic Web summarization techniques.

Description: *AAP - Do Yourself a Favor: Skip the Tan Warns about the effects of suntans, including wrinkles and skin cancer. From the American Academy of Pediatrics.*

Title: AAP - Do Your Skin a Favor: Skip the Spring Break Tan

Meta-Data: *Null*

Figure 1. An example of the human-supplied "good summary": the description, title and meta-data of a page.

4.4.3 Results on unsupervised summarization algorithms

In this section, we evaluate our proposed Web summarization algorithms. We test and compare the content-body identification by page layout analysis, as well as the other two summarization algorithms including "adapted Luhn's algorithm" and "LSA".

As mentioned in Section 3.3, we set a threshold value to determine whether there is a link between the two objects on a Web page. In our experiment, the threshold is set to be 0.1. Through our experiments, we found that most of the unrelated objects in Web pages, such as copyright and advertisement banner, can be easily removed by our algorithm. For example, in Figure 2, the Web page is segmented into four objects by our proposed page layout analysis algorithm. Within these objects, only object 2 (title) and object 3 (main body) are selected as content body; object 1 (banner) and object 4 (copyright) are removed as noisy data. For Luhn's algorithm and LSA algorithm, the compression rate is set as 20% and 30% respectively in our experiments. From Table 3 and Table 4, we found that these three unsupervised summarization algorithms are comparable on classification experiment. All of them can achieve more than 7% improvement as compared to the baseline system.

4.4.4 Result on supervised summarization algorithm

In this experiment, since the Web-page description is authored by Web-directory editors instead of extracted from the Web pages automatically, we need to tag each sentence as positive or negative example for training the supervised summarizer. In our experiment, we define one sentence as positive if it's similarity with the description is greater than a threshold (0.3 in this paper), and others as negative. The F1 measure of the supervised method (denoted by Supervised) is shown in Table 3 and Table 4 (when compression rate equals to 20%). We found it can achieve about 6% relatively improvement compared to baseline system, which is a little worse than unsupervised algorithms. The reason may be that our training data selection is not precise since we only rely on the similarity to descriptions.

4.4.5 Result on Hybrid summarization algorithm

Through the above experiments, we found that both unsupervised and supervised summarization algorithms can improve the classification accuracy to some extent. But none of them can approach the upper bound of the system set by classification by human edited summary. Therefore, in this experiment we are investigating the relative merits of these summarization techniques for classification, and compare with an ensemble of them. From Table 3 and Table 4, we found that all of the summarization algorithms were complementary. The ensemble of summarization methods can achieve about an impressive 12.9% improvement as compared to baseline system, which is also very near to the upper bound of the system. In this experiment, we use the same weighting for each summarization algorithm. We will consider the different weighting schema in the later Section.

4.4.6 Performance on Different Compression Rates

In order to find the relationship between the performance of classification and the compression rate of summarization, we conducted the experiments and the results are shown in Table 5 and Table 6.

Table 5. Performance of CB with different Threshold with NB

	0.20	**0.15**	**0.10**	**0.05**
Content Body	65.0±0.5	67.0±0.4	**69.2±0.4**	66.7±0.3

Table 6. Performance on different compression rate with NB

	10%	**20%**	**30%**	**50%**
Luhn	66.1±0.5	**69.8±0.5**	67.4±0.4	64.5±0.3
LSA	66.3±0.6	67.0±0.5	**68.1±0.5**	63.4±0.3
Supervised	66.1±0.5	**67.3±0.4**	64.8±0.4	62.9±0.3
Hybrid	66.9±0.4	69.3±0.4	**71.8±0.3**	67.1±0.3

From Table 5 and Table 6 we found that all the methods reach their peak performance when the compression rate is 20% or 30% (for CB when the threshold equals to 0.10). However, when the compression rate rises to 50%, the performance of some methods such as LSA and supervised summarization become worse than the baseline. This may be ascribed to the inclusion of noises with the raise of the compression rate.

4.4.7 Effect of different weighting schemata

In the section, experiments are conducted to test the effect of different weighting schema. We tested five cases denoted by Schemas 1—5 which assigns different weights to different summarization scores, in addition to the original schema which sets an equal weight for each summarization algorithm. For simplicity, we modified the equation in Section 3.5 as

$$S = w_1 S_{luhn} + w_2 S_{lsa} + w_3 S_{cb} + w_4 S_{\sup}$$

Schema1: We assigned the weight of each summarization method in proportion to the performance of each method (the value of micro-F1).

Schema2-5: We increased the value of w_i (i=1, 2, 3, 4) to 2 in Schema2-5 respectively and kept others as one.

From the results shown in Table 7, we can conclude that different schemata made no obvious difference.

Table 7. Effect of different weighting schema with NB

	microP	microR	micro-F1
Origin	80.2±0.3	65.0±0.3	71.8±0.3
Schema1	81.0±0.3	65.6±0.3	72.5±0.3
Schema2	**81.3±0.4**	**66.1±0.4**	**72.9±0.4**
Schema3	79.5±0.4	64.4±0.4	71.2±0.4
Schema4	81.1±0.3	65.5±0.3	72.5±0.3
Schema5	79.7±0.4	64.7±0.4	71.4±0.4

4.5 Case studies

In the experiments above, we observed that all the summarization methods achieve some clear improvement as compared to the baseline by either NB or SVM classifier. In order to find out the underlying reasons why summarization can help the classification, in this section we conduct a further case study experiment.

We randomly selected 100 Web pages that are correctly labeled by all our summarization based approaches but wrongly labeled by the baseline system (denoted as set A) and 500 pages randomly from the testing pages (denoted as set B). We find that the average size of pages in A which is 31.2k is much larger than that in B which is 10.9k. The difference shows that the useful information about the content of Web pages is more likely to be missing in larger-sized pages. The summarization techniques can help us to extract this useful information from the large pages.

To illustrate, we show a relatively simple Web page (http://www.aap.org/advocacy/releases/safeskin.htm) from A to show how our approaches work. This example page is shown in Figure 2. The summary given by the human editor including the description, title and meta-data is shown in Figure 1. As we can see, the description is very clear and indeed captures the Web page's main topic without introducing noise. Thus the performance based on this summary is always the best one. However the meta-data for this page is empty, which accounts for the poor performance of classification based on "pure meta-data".

By analyzing the layout of the page, we can separate it into four objects as shown in Figure 2. Objects 2 and 3 were extracted as Content Body, which correspond nicely human intuition since objects 1 and 4 were not that related to the topic of the page.

The summaries created by Luhn's method, LSA and supervised method are not shown in this paper due to space limitation. We found that most of the sentences selected by the above summarization method are correctly included in the summary. Though the supervised method itself may introduce some noise, the ensemble-based method can successfully rule out the noise.

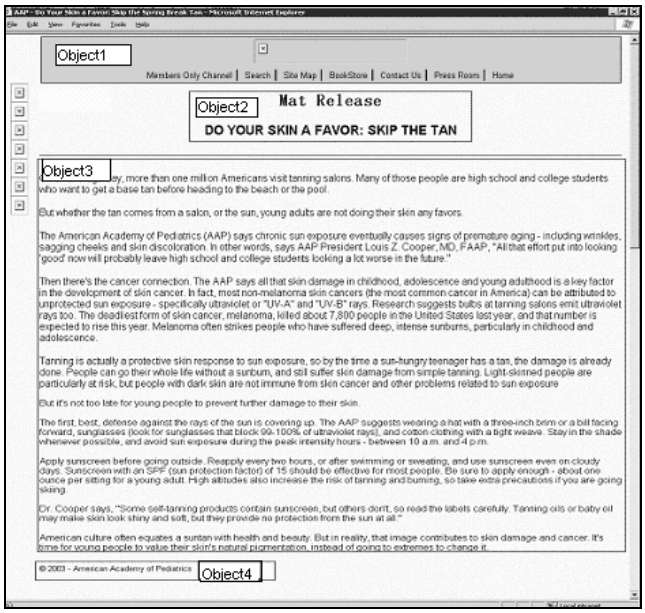

Figure 2. An example to illustrate our approaches

5. CONCLUSIONS AND FUTURE WORK

In this paper, several Web-page summarization algorithms are proposed for extracting the most relevant features from Web pages for improving the accuracy of Web classification. As illustrated by our ideal-case experiment, the summary created by human editors can achieve more than 13.2% improvement by the micro-F1 measure as compared to the pure text of the Web pages. This observation validates the need to find better Web-page summarization methods. We evaluated Web-page categorization on several state-of-the-art automatic document summarization algorithms, as well as an algorithm by utilizing the layout analysis of Web pages. Experimental results show that automatic summary can achieve a similar improvement (about 12.9% improvement) as the ideal-case accuracy achieved by using the summary created by human editors.

In this paper, we only considered a Web page as an isolated document. However, more and more research works demonstrate that the hyperlink is one of the important features for Web search and analysis. In the future, we will investigate methods for multi-document summarization of the hyperlinked Web pages to boost the accuracy of Web classification.

6. REFERENCES

[1] G. Attardi, A. Gulli, and F. Sebastiani. Automatic Web Page Categorization by Link and Context Analysis. In Chris Hutchison and Gaetano Lanzarone (eds.), Proc. of THAI'99, 1999, 105-119.

[2] A.L. Berger, V.O. Mittal. OCELOT: A System for Summarizing Web Pages. Proc. of the 23rd annual international ACM SIGIR, Athens, Greece, 2000, 144-151.

[3] M.W. Berry, S.T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. SIAM Review, 37:573-595, 1995.

[4] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: text summarization for Web browsing on handheld devices. Proc. of WWW10, Hong Kong, China, May 2001.

[5] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. Proc. of the ACM SIGMOD, 1998.

[6] H. Chen and S. T. Dumais. Bringing order to the Web: Automatically categorizing search results. Proc. of CHI2000, 2000, 145-152.

[7] J.L. Chen, B.Y. Zhou, J. Shi, H.J. Zhang, and Q.F. Wu. Function-based Object Model Towards Website Adaptation, Proc. of WWW10, HK, China, 2001.

[8] Z. Chen, S.P. Liu, W.Y. Liu, G.G. Pu, W.Y. Ma. Building a Web Thesaurus from Web Link Structure. Proc. of the 26th annual international ACM SIGIR, Canada, 2003, 48 - 55.

[9] W. Chuang , J. Yang, Extracting sentence segments for text summarization: a machine learning approach, Proc. of the 23rd annual international ACM SIGIR, Athens, Greece, 2000, 152-159

[10] C. Cortes and V. Vapnik. Support vector networks. Machine Learning, 20:1-25, 1995.

[11] S. Deerwester, S. Dumais, G. Furnas, T. Landauer,and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, vol. 41, 1990, 391-407.

[12] J.-Y. Delort, B. Bouchon-Meunier and M. Rifqi. Web Document Summarization by Context. Poster Proc. of WWW12, 2003.

[13] E. J. Glover, K. Tsioutsiouliklis, and et al. Flake. Using Web structure for classifying and describing Web pages. Proc. of WWW12, 2002.

[14] Y.H. Gong, X. Liu. Generic text summarization using relevance measure and latent semantic analysis. In Proc. Of the 24th annual international ACM SIGIR, New Orleans, Louisiana, United States, 2001, 19 - 25.

[15] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In Proceedings of ECML-98, 10th European Conference on Machine Learning, 1998, 137-142.

[16] T. Joachims. Transductive inference for text classification using support vector machines. Proc. of ICML-99, Bled, Slovenia, June 1999.

[17] S.J. Ker and J.-N. Chen. A Text Categorization Based on Summarization Technique. In the 38th Annual Meeting of the Association for Computational Linguistics IR&NLP workshop, Hong Kong, October 3-8, 2000.

[18] Y.J Ko, J.W Park, J.Y. Seo. Automatic Text Categorization using the Importance of Sentences. Proc. of COLING 2002.

[19] A. Kolcz, V. Prabakarmurthi, J.K. Kalita. Summarization as feature selection for text categorization. Proc. Of CIKM01, 2001.

[20] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. Proc. of the 18th annual international ACM SIGIR, United States, 1995, 68-73.

[21] W. Lam, Y.q. Han. Automatic Textual Document Categorization Based on Generalized Instance Sets and a Metamodel. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5): 628-633,2003

[22] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to Latent Semantic Analysis. Discourse processes, 25, 1998, 259-284.

[23] H.P. Luhn. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, Vol. 2, No. 2, April 1958, 159-165.

[24] A. McCallum and K. Nigam, A comparison of event models for naive bayes text classification, In AAAI-98 Workshop on Learning for Text Categorization, 1998.

[25] T. Mitchell. Machine Learning. McGraw-Hill, 1997.

[26] W. Press and et al., Numerical Recipes in C: The Art of Scientific Computing. Cambridge, England: Cambridge University Press, 2 ed., 1992.

[27] F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, , 2002.

[28] Sequential Minimal Optimization, http://research.micro-soft. com/~jplatt/smo.html.

[29] The Porter Stemming Algorithm, http://www.tartarus.org/~martin/PorterStemmer.

[30] S. Teufel and M. Moens. Sentence extraction as a classification task. In ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization, 1997.

[31] C. J. van Rijsbergen. Information Retrieval. Butterworth, London, 1979, 173-176.

[32] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, NY, USA, 1995.

[33] L. Yi, B. Liu, and X. Li. Eliminating Noisy Information in Web Pages for Data Mining. KDD2003. 2003.

[34] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. Proc. of ICML-97.