Title Extraction from Bodies of HTML Documents and its Application to Web Page Retrieval

Yunhua Hu [1], Guomao Xin [2], Ruihua Song, Guoping Hu [3],
Shuming Shi, Yunbo Cao, and Hang Li

Microsoft Research Asia, Beijing China
1 Xi'an Jiaotong University, Xi'an China
2 Peking University, Beijing China
3 University of Science and Technology of China, Hefei China

Contact: hangli@microsoft.com

ABSTRACT

This paper is concerned with automatic extraction of titles from the bodies of HTML documents. Titles of HTML documents should be correctly defined in the title fields; however, in reality HTML titles are often bogus. It is desirable to conduct automatic extraction of titles from the bodies of HTML documents. This is an issue which does not seem to have been investigated previously. In this paper, we take a supervised machine learning approach to address the problem. We propose a specification on HTML titles. We utilize format information such as font size, position, and font weight as features in title extraction. Our method significantly outperforms the baseline method of using the lines in largest font size as title (20.9%-32.6% improvement in F1 score). As application, we consider web page retrieval. We use the TREC Web Track data for evaluation. We propose a new method for HTML documents retrieval using extracted titles. Experimental results indicate that the use of both extracted titles and title fields is almost always better than the use of title fields alone; the use of extracted titles is particularly helpful in the task of named page finding (23.1% -29.0% improvements).

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search Process*; D.2.8 [**Software Engineering**]: Metrics - *complexity measures, performance measures*

General Terms

Algorithms, Experimentation, Performance

Keywords

Information Retrieval, HTML Document, Metadata Extraction

SIGIR'05, August 15-19, 2005, Salvador, Brazil.

1. INTRODUCTION

In this paper we address the issue of automatically extracting titles from the bodies of HTML documents.

Titles are the 'names' of documents and thus are very useful information for document processing. In HTML documents, authors can explicitly specify the title fields marked by '<title>' and '</title>'.. However, usually people do not do it carefully. We have evaluated the title fields of HTML documents in the TREC Web Track data set containing 1,053,111 HTML documents. We have found that about 33.5% of the title fields are somewhat bogus (see Section 6 for details).

In fact, 'true' titles also exist in the bodies of HTML documents. They tend to be more reliable, because they are more noticeable to readers and thus usually are more carefully created by the authors. One question arises here: can we extract titles from the bodies and use them in web applications? We address the problem in this paper.

To the best of our knowledge, no previous work has been conducted on exactly this problem. Title extraction from the bodies of HTML documents is not easy as it appears to be. There is much variability in the format and content of web pages. It was not clear whether it would be possible to conduct the extraction and whether it is possible to use the extracted data in an application.

The key issues to the task are to define a specification of HTML titles and to define features for the extraction.

We take a machine learning approach to address the problem. We propose a specification on HTML titles. The specification takes the 'most conspicuous' description in the document as a title and is defined mainly based on format information. We annotate titles in sample documents and take them as training data, train a classification model, and perform title extraction using the model. In the model, we mainly utilize format information such as font size, position, font weight as features. There are in total 245 features defined. As classification model, we employ Perceptron with Uneven Margins.

We also propose a new retrieval method for web page retrieval. The Okapi-based method combines text, title, and extracted title. It conducts normalization on each type of score calculated with each type of data, and combines the normalized scores using linear combination.

Experimental results indicate that for HTML title extraction our method can significantly outperform the baseline: one that always

uses the lines in the largest font sizes as titles (20.9%-32.6% improvement in F1 score).

Experimental results, on the TREC Web Track data, also indicate that the use of both extracted titles and title fields is almost always better than the use of title fields alone. Moreover, the use of extracted titles is particularly helpful in the task of named page finding in TREC (23.1% -29.0% improvements).

Other empirical findings include that font size is the most important feature for HTML title extraction. A model trained in one domain can be applied to another domain with nearly no drop in accuracy.

The rest of the paper is organized as follows. In Section 2, we introduce related work, and in Section 3, we explain the motivation and setting of our work. In Section 4, we describe our method of title extraction and in Section 5, we describe our method of web page retrieval using extracted titles. Section 6 gives our experimental results. We make concluding remarks in Section 7.

2. RELATED WORK

Web information extraction has become a popular research area recently and many issues have been intensively investigated [10].

Automatic extraction of web information has been studied for different information types. For instance, Liu et al. proposed a method of extracting data records from web pages [16]. Reis et al. investigated the issue of extracting news articles [20]. Craven proposed a method of extracting summaries from web pages [6].

Web information extraction has also been conducted with different data units. For instance, Breuel has proposed parsing web pages as trees of HTML tags (called the DOM tree) and pulling out information from the trees [2] (See also [14, 20]). Song et al. have proposed to divide web pages into a number of blocks and conduct information extraction based on the blocks [22]. Specifically, they have developed a method of identifying important blocks in a page.

Web information extraction can be either domain specific or domain independent. In a specific domain, one can assume that the structures of web pages are similar and the similar structures can be used in learning and extraction [7, 14]. For instance, Kosala et al. have observed that news articles at a web site usually share the same template and they have attempted to perform information extraction from news articles using the template [14].

In general, there are two approaches to web information extraction: namely, the rule based approach and the machine learning based approach. The machine learning based approach is more widely employed [3, 6, 7, 8, 10, 11, 12, 13, 15, 16, 17, 20, 22, 25].

To the best of our knowledge, there has been no previous work on title extraction from HTML body, particularly on open domain and by using format information. Our work is close to [2, 20, 25] in the sense that we also use DOM tree, but it is also different in that we target title extraction. Our work also differs from the work on block-based extraction [22]. Important block identification may help title extraction, but not in a straightforward way (e.g., a title may exist in a small and 'unimportant' block).

In information retrieval, previous work has shown that the use of title fields, anchor texts, and URLs of web pages (HTML documents) can enhance web page retrieval. Cutler et al. have proposed using the structures in HTML documents to improve HTML document retrieval [9]. Specifically, they linearly combine term frequencies in several fields extracted from an HTML document. In TREC-2002, several participants [1, 4, 25] have reported that they utilize different fields in HTML files for web page retrieval. Zhang et al. have explored the roles of different HTML fields such as title field, bold text, etc. in ranking [25, 26]. Amitay et al. have proposed eliminating documents whose title fields do not contain any query word in document retrieval [1]. In TREC-2003, more than half of the participants have considered the use of richer representations/surrogates based on document structures [5]. For instance, Ogilvie and Callan have tried to make use of different document representations from different sources in language models for the named page and homepage finding tasks [18, 19]. See also [24].

It has been not clear previously whether and how extracted titles can be used for enhancing web page retrieval.

3. PROBLEM

We consider the problem of automatically extracting titles from the bodies of HTML documents (web pages). We assume, in this paper, that we conduct title extraction on the general domain and thus we can only utilize domain independent information (mainly format information) in the extraction.

We first consider what can be defined as titles of HTML documents. We give a 'specification' on the titles. The specification defines titles mainly from the viewpoint of document format. Intuitively, a title of an HTML document is the 'most conspicuous' description in the document.

In the example in figure 1, one title is "National Weather Service Oxnard" and the other title is "Los Angeles Marine Weather Statement". In the example in figure 2, one title is "State Assembly – District 34" and the other title is "1998 California General Election Certified List of Candidates".

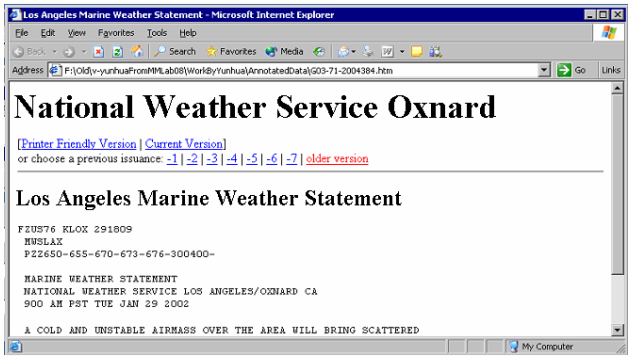

Figure 1. Example web page.

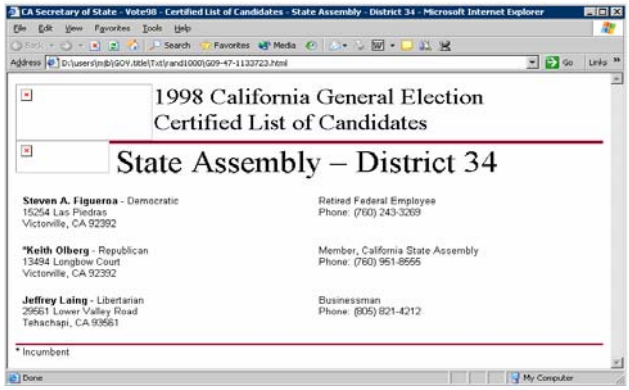

Figure 2. Example web page.

The specification is as follows.

1. **Number**
   An HTML document can have two titles, one title, or no title.
2. **Position**
   a) Titles must be on the top region (i.e., within the top ¼ region);
   b) Titles cannot be in a narrow pane (i.e., with a width of less than ¼ of the entire page).
3. **Appearance**
   a) The font sizes of titles are usually the largest and second largest;
   b) Titles are conspicuous in terms of font family, font weight, font color, font style, alignment, background color, and text length.
4. **Neighbor**
   a) Titles consist of consecutive lines in the same *font size*. (i.e., subtitles in smaller font sizes should be ignored);
   b) Titles cannot be a part of bullets or numbering, and cannot be chapter or section names;
   c) If there exist two titles, then the two titles usually are in two different blocks. Lines, links, and images can be separators between the blocks.
5. **Content**
   a) Titles cannot be a link, time expression, address, etc;
   b) Titles cannot be 'under construction', 'last updated', etc;
   c) Titles can be the expressions immediately after 'Title:' and 'Subject:'.
6. **Other**
   a) Titles in images are not considered.

4. TITLE EXTRACTION METHOD

4.1 Process

In this paper, we take a machine learning approach to address the problem. Our method consists of two phases: training and extraction. There is the same pre-processing for training and extraction. There is also a post-processing for extraction.

The input of preprocessing is a document. In the pre-processing, we extract units from the input document. The output of pre-processing is a sequence of units (instances). A unit generally corresponds to a line in the HTML document. It contains not only content information (linguistic information) but also format information. Figure 3 shows the units obtained from an HTML document. Specifically, we parse the body of the HTML document and construct a DOM (Document Object Model) tree. Figure 4 shows an example DOM tree. We take all the leave nodes that contain 'texts' in the DOM tree as units.

```
Unit 1:
Unit 2: [text="Microsoft Corporation", alignment=center,boldface=false,italic=false,
isH1=false,largest-font=false,second_font_size=false...]
Unit 3:
Unit 4: [text="Windows Operating System",
alignment=center,boldface=false,italic=false, isH1=true,largest_font_size=false,...]
Unit 5: [text="Overview",alignment=left,boldface=false,italic=true,isH1=false,
largest_font_size=false,second_font_size=true...]
...
```

Figure 3. Units in HTML documents.

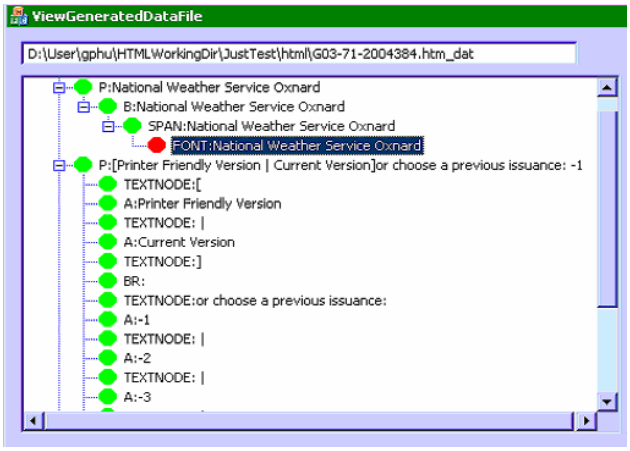

Figure 4. The DOM tree of the document in figure 1.

In learning, the input is sequences of units, and each sequence corresponds to one document. We take labeled units (titles and others) in the sequences as training data and construct a model for identifying whether a unit is a title.

In extraction, the input is a sequence of units from one document. We employ the model to identify each unit in the sequence to find whether it is a title. The model assigns a score to each unit.

In post-processing of extraction, we extract titles using heuristics. The output is the extracted titles of the document. Specifically, we choose the consecutive units with the highest scores as the first title, and then choose the consecutive units with the second highest scores as the second title, provided that the scores are larger than zero.

4.2 Model

We describe the model in a general framework. The input is sequences of instances $x_1 \cdots x_n$ with aligned sequences of labels $y_1 \cdots y_n$. Instances represent original data, i.e., units. Labels represent being or not being the target of extraction, i.e., title. Suppose that $X_1 \cdots X_n$ are random variables denoting a sequence of instances, and $Y_1 \cdots Y_n$ are random variables denoting a sequence of labels. If the *Y*s are independent from each other, then we have

$$P(Y_1 \cdots Y_n \mid X_1 \cdots X_n) = P(Y_1 \mid X_1 \cdots X_n) \cdots P(Y_k \mid X_1 \cdots X_n)$$

Each conditional probability model is a classifier. In this paper, as classifier we employ an improved variant of Perceptron, called Perceptron with Uneven Margin [15]. This version of Perceptron can work well especially when the number of positive training instances and the number of negative training instances differ largely, which is exactly the case for the current problem.

4.3 Features

The main characteristic of our method is that we mostly utilize format information for extraction. Title extraction mainly based on format information is not an easy task, because the formats (e.g., layouts) of web pages (HTML documents) can vary largely.

We manage to use as many effective features as possible.

We consider the use of the following information in the design of features.

1. **Rich format information**
   a) Font size: 1~7 levels
   b) Font weight: bold face or not
   c) Font family: Times New Roman, Arial, etc
   d) Font style: normal or italic
   e) Font color: #000000, #FF0000, etc
   f) Background color: #FFFFFF, #FF0000, etc
   g) Alignment: center, left, right, and justify.
2. **Tag information**
   a) H1,H2,…,H6: levels as header
   b) LI: a listed item
   c) DIR: a directory list
   d) A: a link or anchor
   e) U: an underline
   f) BR: a line break
   g) HR: a horizontal ruler
   h) IMG: an image
   i) Class name: 'sectionheader', 'title', 'titling',' header', etc.
3. **Position information**
   a) Position from the beginning of body
   b) Width of unit in page
4. **DOM tree information**
   a) Number of sibling nodes in the DOM tree.
   b) Relations with the root node, parent node and sibling nodes in terms of font size change, etc.
   c) Relations with the previous leaf node and next leaf node, in terms of font size change, etc. Note that the nodes might not be siblings.
5. **Linguistic information**
   a) Length of text: number of characters
   b) Length of real text: number of alphabet letters
   c) Negative words: 'by', 'date', 'phone', 'fax', 'email', 'author', etc.
   d) Positive words: 'abstract', 'introduction', 'summary', 'overview', 'subject', 'title', etc.

With the information above, we create four types of features which can help identify the position (Pos), appearance (App), neighbor (Nei), and content (Con) of a title. There are in total 245 binary features. Table 1 gives example features for each type. Table 2 shows the number of features for each type.

Table 1. Examples of features

Type	Feature descriptions
Pos	Unit in top 0.2, top 0.4, or rest of page
Pos	Unit width < 0.1, 0.2, 0.3, or 0.4 of page width
Pos	First unit in DOM tree
App	Is unit tagged with H1,…, H6, or no H* tag
App	Is first, second, or third H* tagged unit
App	Is top or second level H* tagged unit in DOM tree
App	Is only H* tagged unit in DOM tree
App	Largest, or second largest font size
App	Percentage of unit font size <0.02, 0.02~0.10, etc
App	Alignment of unit is center, left, right, or justify
App	Is unit in bold face
App	Unit is italic
App	Unit is underlined
App	Percentage of units in same color is <0.05, 0.05~0.21, etc
App	Percentage of units in same background color is <0.05, 0.05~0.21, etc
App	Percentage of units in same font family is <0.05, 0.05~0.21, etc
App	In bullet (i.e., tagged with LI)
App	Begin with newline
Con	All characters, or first characters capitalized
Con	Number of characters is <8, 8-64, 64-96, >96
Con	Begins with "subject:", "introduction", "title", "overview", etc
Nei	Previous or next unit is tagged as HR, BR, etc
Nei	Font size is larger than root, node, previous leaf, next leaf, or brother node
Nei	Alignment of previous node is left and current is center, etc
Nei	If the same level units have same font size, font weight, etc

Table 2. Distribution of feature types

Type	Number
Pos	8
App	97
Nei	100
Con	15
Other	25

5. DOCUMENT RETRIEVAL METHOD

We propose a linear combination method for using extracted titles in document retrieval. Our method takes BM25 as basic function and is unique in its way of normalizing the BM25 scores.

Given an HTML document, we extract information from it and store the result in several fields: body, title, and extracted title. The extracted title field contains the title extracted by our method. We also create an additional field in which we combine the extracted title field and the title field. We denote it as 'CombTitle'. We consider four methods for document retrieval with different uses of the fields.

BasicField

In this method, a document is represented by all the texts in the title and body. Given a query, we employ BM25 to calculate the score of each document with respect to the query:

$$S = \sum_{i \in q} \frac{(k_1+1)tf_i}{k_1((1-b)+b\frac{dl}{avdl})+tf_i} \log \frac{N-df_i+0.5}{df_i+0.5} \qquad (1)$$

Here, i denotes a word in the query q; tf_i and df_i are term frequency and document frequency of i respectively; dl is document length, and $avdl$ is average document length; k_1 and b are parameters. We set $k_1 = 1.1, b = 0.7$.

BasicField+CombTitle

We calculate the BM25 score of the combined field CombTitle (i.e., view it as a document), with $k_1 = 0.4, b = 0.95$. We also calculate the BM25 score of BasicField as in the baseline method. We next conduct normalization on both the BM25 score of the combined field and that of the baseline method.

$$S' = \frac{\left(\sum_{i \in q} \frac{(k_1+1)tf_i}{k_1((1-b)+b\frac{dl}{avdl})+tf_i} \log \frac{N-df_i+0.5}{df_i+0.5} \right)}{\left(\sum_{i \in q} (k_1+1) \log \frac{N-df_i+0.5}{df_i+0.5} \right)} \quad (2)$$

We next linearly combine the two scores, $S'_{\text{BasicFields}}$ and S'_{ComTitle},:

$$\alpha S'_{\text{BasicField}} + (1-\alpha) S'_{\text{ComTitle}} \quad (3)$$

Here α is coefficient ranging from 0 to 1.

BasicField+ExtTitle

We employ a similar method to that of BasicField+ComTitle, in which instead of using the combined title filed, we use the extracted title field.

BasicField+Title

This is a similar method to BasicField+ComTitle, in which instead of using the combined title filed, we use the title field.

6. EXPERIMENTAL RESULTS

6.1 Data Sets

As data sets, we used the .GOV data in the TREC Web Track and the data from an intranet of Microsoft. We call the former TREC and the latter MS, hereafter. There are 1,053,111 web pages in TREC and about 1,000,000 web pages in MS.

We randomly selected 4,258 and 4,137 HTML documents from the two data sets respectively. We manually annotated titles in the randomly selected HTML documents. The annotation was based on the specification in Section 3. There were 3,332 HTML documents with annotated titles in TREC, and there were 2,641 HTML documents with annotated titles in MS (Recall that an HTML document can have no title).

6.2 Evaluation Measures for Extraction

We used 'precision', 'recall', 'F1-score', and 'accuracy' in evaluation of title extraction results. In the evaluation, if the extracted title can approximately match to the annotated title, then we view it as a correct extraction. We define the approximate match between the two titles *t1* and *t2* in the following way.

$$\frac{d(t1,t2)}{\max(l1,l2)} < 0.3$$

where *d(t1,t2)* is the edit distance between *t1* and *t2*; *l1* and *l2* are lengths of *t1* and *t2* respectively.

6.3 Evaluation of Title Fields

We tried to see how many title fields in the HTML documents are correct. We found that there are 33.5% of HTML documents in the TREC data set having bogus titles. There are three cases:

1. Empty title field

There are 60,524 pages (5.8%) which have nothing in the title fields, i.e., empty between '<TITLE>' and '</TITLE>'.

2. 'Untitled' title field

There are 4,964 pages (0.8%) which have "untitled" or "untitled document" in their title fields.

3. Duplicated title field

282,826 pages (26.9%) fall into this type. Many web sites contain web pages sharing the same title field but having different contents. In our investigation, a title filed is considered duplicated if it is repeated more than *N* times in a web site. *N* is determined heuristically on the basis of *M* the total number of pages in the web site. There are five rules:

if $M > 1000$ then $N = 1/40 \cdot M$; …; if $M < 60$ then $N = 5$.

6.4 Title Extraction Experiment

We conducted title extraction experiments on the two data sets.

For our method (denoted as Perceptron for short), we conducted 5-fold cross validation, and thus all the results are averaged over 5 trials.

As baseline methods, we used that of always extracting the largest font size units and that of always extracting the first unit. We also evaluated the titles in title fields denoted as 'title-field'.

Tables 3 and 4 show the results. The results indicate that our method significantly outperforms the baseline methods and title-field. It seems that the use of only one type of information for extraction is not enough. Our learning-based method can make an effective use of various types of information in title extraction.

Table 3. Performances of title extraction methods on TREC

Approach	Precision	Recall	F1-Score	Accuracy
Largest font (Baseline)	0.528	0.643	0.580	0.523
First unit	0.327 (-38.1%)	0.402 (-37.5%)	0.360 (-37.8%)	0.327 (-37.5%)
Title-field	0.270 (-48.8%)	0.324 (-49.6%)	0.295 (-49.1%)	0.261 (-50.0%)
Perceptron	**0.698 (+32.3%)**	**0.703 (+9.3%)**	**0.701 (+20.9%)**	**0.698 (+33.5%)**

Table 4. Performances of title extraction methods on MS

Approach	Precision	Recall	F1-Score	Accuracy
Largest font (Baseline)	0.584	0.840	0.689	0.582
First unit	0.606 (+3.7%)	0.875 (+4.1%)	0.716 (+3.9%)	0.606 (+4.1%)
Title-field	0.656 (+12.3%)	0.834 (-0.7%)	0.735 (+6.6%)	0.673 (+15.6%)
Perceptron	**0.910 (+55.7%)**	**0.919 (+9.4%)**	**0.914 (+32.6%)**	**0.909 (+56.1%)**

The performance of title extraction in MS is better than that in TREC. We found that the HTML documents in MS have fewer patterns than those in TREC and that is the reason for the higher performance.

We further investigated the relation between the titles in title fields and the titles extracted from the bodies. Table 5 shows the results. We see that our method can still achieve a relatively high performance when title-fields are incorrect. In this case, the extracted titles are particularly useful. We also see that when title-fields are correct, we have better performance, but still cannot conduct the extraction completely correctly. The result indicates that the task of title extraction is not easy.

Table 5. Performances of title extraction with respect to different accuracies of title-field

Data Set	Title-field is incorrect	Title-field is correct
TREC	0.672	0.737
MS	0.855	0.952

6.5 Web Retrieval Experiment

We conducted web page retrieval experiments on the TREC data.

In the experiment, we used the queries and relevance judgments of Web Tracks in TREC-2002, TREC-2003, and TREC-2004. The queries have been classified into three types, i.e. named-page finding (NP), homepage finding (HP) and topic distillation (TD) [5]. The number of queries in each type for each year is listed in table 6. The topic distillation queries of TREC-2002 were not used because the specification in it is different from those in TREC-2003 and TREC-2004.

Table 6. Distribution of queries

Year	Task	Number of queries
2002	NP	150
2003	TD	50
	NP + HP	150 + 150
2004	TD + NP + HP	75 + 75 + 75

We first performed the experiment using the queries of TREC-2003. We applied the three methods BaseFiled+Title, BaseField+CombTitle, BaseFiled+ExtTitle to the TREC data and evaluated the results in terms of Mean Average Precision. Figures 5, 6, and 7 show how the performances of the three methods change when the coefficient of alpha changes, in the three tasks: NP, HP, and TD. The baseline method is BaseField and its performance is obtained when alpha equals 1. The results indicate that the best of BaseField+CombTitle outperforms the baseline method in all three tasks. This is also true for BaseField+Title. However, BaseField+ExtTitle can only beat the baseline for NP and TD, but cannot beat the baseline for HP. Furthermore, BaseFiled+CombTitle is always better than BaseFiled+Title. The results indicate that the extracted titles are useful for web page retrieval, especially for NP and it is better to employ BaseFiled+CombTitle.

We also note that when alpha equals 0, all the three methods turn out to be one without using BaseField. For all the three tasks, the performances of CombTitle alone are close to or better than the baseline. The result implies that CombTitle, a combination of title and extracted title, can serve as a good summarization of an HTML document.

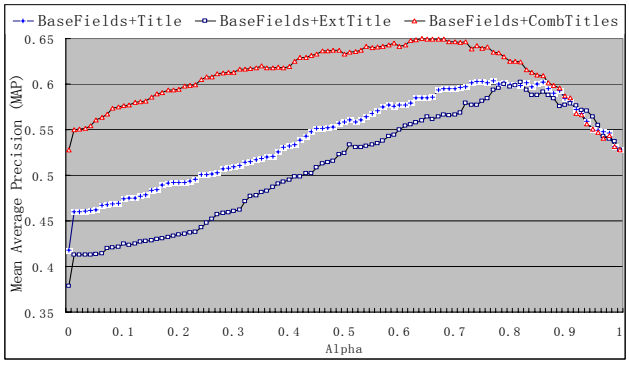

Figure 5. Web retrieval results with TREC-2003 NP.

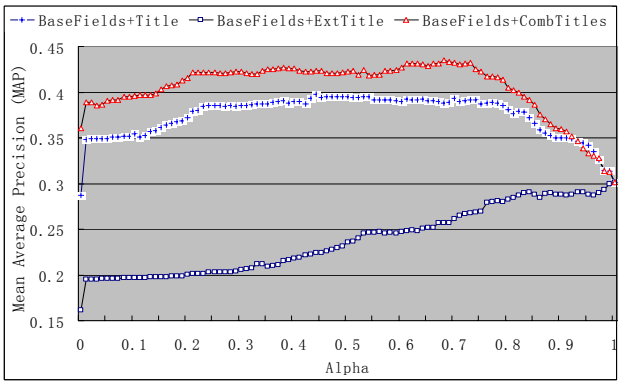

Figure 6. Web retrieval results with TREC-2003 HP.

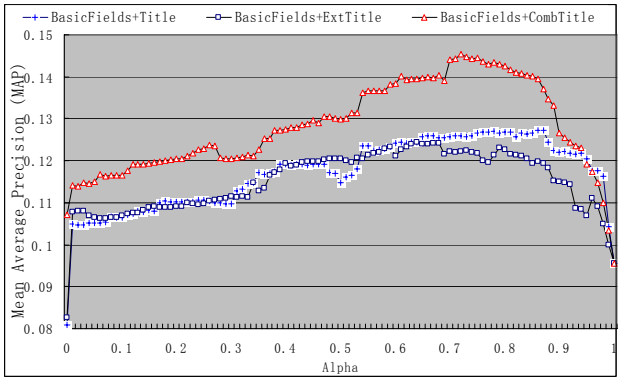

Figure 7. Web retrieval results with 2003 TD.

Table 7 shows the best result for each method in each task. The results with the ">>" mark are those significantly better than the baseline of BasicField in 'T-test'. For NP, the improvement of BasicField+CombTitle over BasicField is statistically significant, but the improvement of BasicField+Title over BasicField is not. For HP and TD, the improvements of both BasicField+Title and BasicField+CombTitle over BasicField are not statistically significant.

We next conducted web retrieval experiments on the data of TREC-2002 and TREC-2004, using the optimal values of alpha obtained from TREC-2003. Table 8 shows the result. Again, for NP BasicField+CombTitle significantly outperforms BasicField and outperforms BasicField+Title. For HP, although the improvement of BasicField+CombTitle over BasicField is significant, the improvement is smaller than that of BasicField+Title, indicating that the use of extracted titles does not help HP. Moreover, the improvements of both BasicField+Title and BasicField+CombTitle are small for TD.

Table 7. Best retrieval results (average precision) on TREC-2003

	BasicField	+Title	+CombTitle
2003.NP	0.528	0.604 (+14.4%)	0.650 (>>) (+23.1%)
2003.HP	0.302	0.397 (>>) (+31.4%)	0.435 (>>) (+44.0%)
2003.TD	0.096	0.127 (+32.3%)	0.145 (+51.0%)

Table 8. Retrieval results (average precision) on TREC-2002 and TREC-2004

	BasicField	+Title	+CombTitle
2004.NP	0.488	0.574 (+17.6%)	0.630 (>>) (+29.0%)
2004.HP	0.272	0.457 (>>) (+68.0)	0.415 (>>) (+52.5%)
2004.TD	0.098	0.106 (+8.2)	0.109 (+11.2%)
2002.NP	0.593	0.587 (-1.0%)	0.647 (>>) (+9.1%)

Finally, we combined BasicField+CombTitle and the use of anchor text and URL. We conducted an experiment on TREC-2004 data. Table 9 shows the final result [23].

We conclude that extracted titles are useful for web page retrieval.

Table 9. Final retrieval results on TREC-2004

Run		Our method
TD	P@10	0.251
	MAP	0.178
	R-Pre	0.205
NP	MRR	0.672
HP	MRR	0.725
Overall	S@1	0.533
	S@5	0.800
	S@10	0.880
	AveP	0.518

6.6 Domain Adaptation Experiment

To investigate the ability of domain adaptation of our extraction model, we conducted two experiments. In the first experiment, we applied the model trained with the TREC data to the MS data. In the second experiment, we swapped the training and test data sets. Table 10 shows the results.

Table 10. Domain adaptation of HTML title extraction

Testing Set	Trained Set	Precision	Recall	F1-Score	Accuracy
TREC	MS	0.698	0.615	0.654	0.642
MS	TREC	0.852	0.883	0.867	0.871

From the results, we see that the cross domain performance is close to that of within domain in Section 6.4. Again, the performance in MS is better than that in TREC. The results indicate that the patterns of HTML titles are similar in different domains, and thus it is possible to construct a domain independent model for title extraction.

6.7 Feature Contribution Experiment

We investigated the contribution of each feature type in title extraction. We employed all the 245 features (All), or each of the four types: App, Con, Pos, and Nei to train a model and conduct title extraction. Furthermore, we further categorized App into seven subtypes: Font Size, Font Weight, Color, Alignment, Background Color, Font Style, and Font Family, and employed each subtype in model training and title extraction.

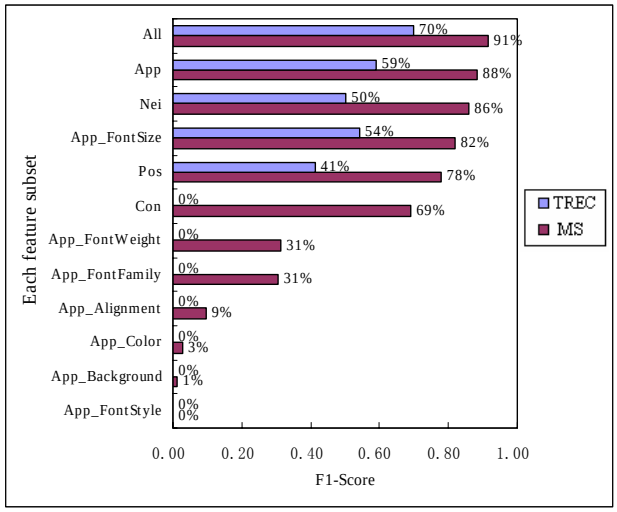

Figure 8. Contribution of each feature type.

Figure 8 shows the F1-score of title extraction with different feature types. We see that the App features are the most significant features, and among the App features, the Font Size features are most important. The results indicate that one type of format information alone is insufficient for accurate title extraction. The results also unveil the reason of the high performance in extraction achieved by our method (in Section 6.4): many types of format information are used.

7. CONCLUSION

In this paper, we have investigated the problem of automatically extracting titles from the bodies of HTML documents, and have investigated how the extracted titles can help improve web page retrieval.

We have proposed a specification of HTML title. We have used a machine learning approach to address the problem. We mainly use format information including rich format, tag, position, and DOM tree information for the extraction.

Our experimental findings include

(1) Our method can work significantly better than the baseline methods for title extraction.

(2) We can construct domain-independent model for title extraction.

(3) Using extracted titles can indeed improve web page retrieval, particularly name page finding of TREC.

(4) Many types of format information are useful for title extraction.

8. ACKNOWLEDGMENTS

We thank Dmitriy Meyerzon, Ming Zhou, and Wei-Ying Ma for their encouragements and supports. We thank Hugo Zaragoza, Nick Craswell and the anonymous reviewers for their comments to this paper.

9. REFERENCES

[1] Amitay, E., Carmel, D., Darlow, A., Lempel, R., and Soffer, A.Topic Distillation with Knowledge Agents, In Proceedings of the Eleventh Text REtrieval Conference (TREC-11), 2002.

[2] Breuel, T.M. Information Extraction from HTML Documents by Structural Matching, In Proceedings of the

Second International Workshop on Web Document Analysis (WDA2003), 2003.

[3] Chidlovskii, B., Ragetli, J., and de Rijke, M. Wrapper Generation via Grammar Induction. In Proceedings of the Eleventh European Conference on Machine Learning (ECML2000), 2000.

[4] Collins-Thompson, K., Ogilvie, P., Zhang, Y., and Callan, J. Information Filtering, Novelty Detection, and Named-Page Findiing, In Proceedings of the Eleventh Text Retrieval Conference (TREC-11), 2002.

[5] Craswell, N. and Hawking, D. Overview of the TREC 2003 Web Track, In Proceedings of the Twelfth Text Retrieval Conference(TREC-2003), 2003.

[6] Craven, T.C. HTML Tags as Extraction Cues for Web Page Description Construction, Informing Science Journal, Volume 6, 2003.

[7] Crescenzi, V., Mecca, G. and Merialdo, P. Roadrunner: Towards Automatic Data Extraction from Large Web Sites. In Proceedings of the Twenty-seventh International Conference on Very Large Databases(VLDB2001), 2001.

[8] Crescenzi, V., Mecca, G. and Merialdo, P. Wrapping-Oriented Classification of Web Pages. In Proceedings of the 2002 ACM Symposium on Applied Computing (SAC-2002), pages 1108-1112, 2002.

[9] Cutler, M., Shih, T. and Meng, Y. Using the Structure of HTML Documents to Improve Retrieval, In Proceedings of the USENIX Symposium on Internet Technologies and Systems (NISTS'97), 1997..

[10] Eikvil, L. Information Extraction from World Wide Web - A Survey. Technical Report 945, 1999.

[11] Evans, D.K., Klavans, J.L. and McKeown, K.R. Columbia Newsblaster: Multilingual News Summarization on the Web. In Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics(HLT-NAACL-2004), 2004.

[12] Freitag, D. Machine Learning for Information Extraction in Informal Domains. Machine Learning, 39(2/3), pages 169-202,2000.

[13] Freitag, D. and McCallum, A. Information Extraction with HMMs and Shrinkage. In Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction (AAAI'99), 1999.

[14] Kosala, R., Bruynooghe, M., Bussche, J.V. and Blockeel, H. Information Extraction from Web Documents Based on Local Unranked Tree Automaton Inference, In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003), 2003.

[15] Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J. and Kandola, J. The Perceptron Algorithm with Uneven Margin. In Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002), 2002.

[16] Liu, B., Grossman, R. and Zhai, Y. Mining Data Records in Web Pages. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD-2003), 2003.

[17] Muslea, I., Minton, S. and Knoblock C. A Hierarchical Approach to Wrapper Induction. In Proceedings of the Third International Conference on Autonomous Agents (Agents'99), 1999.

[18] Ogilvie, P. and Callan, J. Combining Structural Information and the Use of Priors in Mixed Named-Page and Homepage Finding, In Proceedings of the Twelfth Text Retrieval Conference (TREC-12), 2003.

[19] Ogilvie, P. and Callan, J. Combining Document Representations for Known-Item Search. In Proceedings of the Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03), 2003.

[20] Reis, D., Golgher, P., Silva, A. and Laender, A. Automatic Web News Extraction Using Tree Edit Distance. In Proceedings of International WWW Conference (WWW-2004), 2004.

[21] Robertson, S., Zaragoza, H. and Taylor, M. Simple BM25 Extension to Multiple Weighted Fields. In Proceedings of ACM Thirteenth Conference on Information and Knowledge Management(CIKM-2004), 2004.

[22] Song, R., Liu, H., Wen, J.-R. and Ma, W.Y. Learning Block Importance Models for Web Pages, In Proceedings of International WWW Conference (WWW-2004), 2004.

[23] Song, R., Wen, J.-R., Shi, S., Xin, G., Liu, T.-Y., Qin, T., Zheng, X., Zhang, J., Xue, G., and Ma, W.-Y. Microsoft Research Asia at Web Track and Terabyte Track of TREC 2004. In Proceedings of the Thirteenth Text REtrieval Conference Proceedings (TREC-2004), 2004

[24] Yau, H.S. and Hawker, J.S. SA_MetaMatch: Relevant Document Discovery Through Document Metadata and Indexing, In Proceedings of ACM Southeast Regional Conference 2004, 2004.

[25] Zhang, M., Song, R., Lin, C., Ma, L., Jiang, Z., Jin, Y., Liu, Y., Zhao, L. and Ma, S. THU at TREC 2002: novelty, web, and filtering. In Proceedings of the Eleventh Text REtrieval Conference (TREC-11), 2002.

[26] Zhang, M., Song, R. and Ma, S. DF or IDF? On the use of HTML primary feature fields for Web IR. In Proceedings of the Twelfth International World Wide Web Conference (WWW2003), 2003.