Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 42 (2006) 1276–1293
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Automatic extraction of titles from general documents using machine learning

Yunhua Hu [a,*,1], Hang Li [b], Yunbo Cao [b], Li Teng [c], Dmitriy Meyerzon [d], Qinghua Zheng [a]

[a] *Computer Science Department, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shaanxi 710049, China*
[b] *Microsoft Research Asia, 5F Sigma Center, No. 49 Zhichun Road, Haidian, Beijing 100080, China*
[c] *Computer Science and Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China*
[d] *Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA*

Received 9 September 2005; received in revised form 6 December 2005; accepted 7 December 2005
Available online 2 February 2006

Abstract

In this paper, we propose a machine learning approach to title extraction from general documents. By general documents, we mean documents that can belong to any one of a number of specific genres, including presentations, book chapters, technical papers, brochures, reports, and letters. Previously, methods have been proposed mainly for title extraction from research papers. It has not been clear whether it could be possible to conduct automatic title extraction from general documents. As a case study, we consider extraction from Office including Word and PowerPoint. In our approach, we annotate titles in sample documents (for Word and PowerPoint, respectively) and take them as training data, train machine learning models, and perform title extraction using the trained models. Our method is unique in that we mainly utilize formatting information such as font size as features in the models. It turns out that the use of formatting information can lead to quite accurate extraction from general documents. Precision and recall for title extraction from Word are 0.810 and 0.837, respectively, and precision and recall for title extraction from PowerPoint are 0.875 and 0.895, respectively in an experiment on intranet data. Other important new findings in this work include that we can train models in one domain and apply them to other domains, and more surprisingly we can even train models in one language and apply them to other languages. Moreover, we can significantly improve search ranking results in document retrieval by using the extracted titles.

Keywords: Information extraction; Metadata extraction; Machine learning; Search

* Corresponding author. Tel.: +86 29 82 66 52 6321; fax: +86 29 826 63 860.
E-mail addresses: yunhuahu@mail.xjtu.edu.cn (Y. Hu), hangli@microsoft.com (H. Li), yucao@microsoft.com (Y. Cao), lteng@cse.cuhk.edu.hk (L. Teng), dmitriym@microsoft.com (D. Meyerzon), qhzheng@mail.xjtu.edu.cn (Q. Zheng).
[1] The work was conducted when the author was visiting Microsoft Research Asia.

0306-4573/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2005.12.001

1. Introduction

Metadata of documents is useful for many kinds of document processing such as search, browsing, and filtering. Ideally, metadata is defined by the authors of documents and is then used by various systems. However, people seldom define document metadata by themselves, even when they have convenient metadata definition tools (Crystal & Land, 2003). Thus, how to automatically extract metadata from the bodies of documents turns out to be an important research issue.

Methods for performing the task have been proposed. However, the focus was mainly on extraction from research papers. For instance, Han et al. (2003) proposed a machine learning based method to conduct extraction from research papers. They formalized the problem as that of classification and employed Support Vector Machines as the classifier. They mainly used linguistic features in the model.

In this paper, we consider metadata extraction from general documents. By general documents, we mean documents that may belong to any one of a number of specific genres. General documents are more widely available in digital libraries, intranets and the internet, and thus investigation on extraction from them is sorely needed. Research papers usually have well-formed styles and noticeable characteristics. In contrast, the styles of general documents can vary greatly. It has not been clarified whether a machine learning based approach can work well for this task.

There are many types of metadata: title, author, date of creation, etc. As a case study, we consider title extraction in this paper. General documents can be in many different file formats: Microsoft Office, PDF (PS), etc. As a case study, we consider extraction from Office including Word and PowerPoint.

We take a machine learning approach. We annotate titles in sample documents (for Word and PowerPoint, respectively) and take them as training data to train several types of models, and perform title extraction using any one type of the trained models. In the models, we mainly utilize formatting information such as font size as features. We employ the following models: Perceptron with Uneven Margins, Maximum Entropy (ME), Maximum Entropy Markov Model (MEMM), Voted Perceptron Model (VP), and Conditional Random Fields (CRF).

In this paper, we also investigate the following three problems, which did not seem to have been examined previously.

(1) Comparison between models: among the models above, which model performs best for title extraction;
(2) Generality of model: whether it is possible to train a model on one domain and apply it to another domain, and whether it is possible to train a model in one language and apply it to another language;
(3) Usefulness of extracted titles: whether extracted titles can improve document processing such as search.

Experimental results indicate that our approach works well for title extraction from general documents. Our method can significantly outperform the baselines: one that always uses the first lines as titles and the other that always uses the lines in the largest font sizes as titles. Precision and recall for title extraction from Word are 0.810 and 0.837, respectively, and precision and recall for title extraction from PowerPoint are 0.875 and 0.895, respectively. It turns out that the using of *format features* is the key to successful title extraction.

(1) We have observed that Perceptron based models perform better in terms of extraction accuracies. (2) We have empirically verified that the models trained with our approach are generic in the sense that they can be trained on one domain and applied to another, and they can be trained in one language and applied to another. (3) We have found that using the extracted titles we can significantly improve precision of document retrieval (by 10%).

We conclude that we can indeed conduct reliable title extraction from general documents and use the extracted results to improve real applications.

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 problem setting of our work. In Section 4, we describe our method of title extraction, and in Section 5, we describe our method of document retrieval using extracted titles. Section 6 gives our experimental results. We make concluding remarks in Section 7.

2. Related work

2.1. Document metadata extraction

Methods have been proposed for performing automatic metadata extraction from documents; however, the main focus was on extraction from research papers.

The proposed methods fall into two categories: the rule based approach and the machine learning based approach.

Giuffrida, Shek, and Yang (2000), for instance, developed a rule-based system for automatically extracting metadata from research papers in Postscript. They used rules like "titles are usually located on the upper portions of the first pages and they are usually in the largest font sizes". Liddy et al. (2002) and Yilmazel et al. (2004) performed metadata extraction from educational materials using rule-based natural language processing technologies. Mao, Kim, and Thoma (2004) also conducted automatic metadata extraction from research papers using rules on formatting information.

The rule-based approach can achieve high performance. However, it also has disadvantages. It is less adaptive and robust when compared with the machine learning approach.

Han et al. (2003), for instance, conducted metadata extraction with the machine learning approach. They viewed the problem as that of classifying the lines in a document into the categories of metadata and proposed using Support Vector Machines (SVM) as the classifier. They mainly used linguistic information as features. They reported high extraction accuracy from research papers in terms of precision and recall. Peng and McCallum (2004) also conducted information extraction from research papers. They employed Conditional Random Fields (CRF) as model.

2.2. Information extraction

Metadata extraction can be viewed as an application of information extraction, in which given a sequence of instances, we identify a subsequence that represents information in which we are interested. Hidden Markov Model (Ghahramani & Jordan, 1997), Maximum Entropy Model (Berger, Della Pietra, & Della Pietra, 1996; Chieu & Ng, 2002), Maximum Entropy Markov Model (McCallum, Freitag, & Pereira, 2000), Support Vector Machines (Cortes & Vapnik, 1995), Conditional Random Field (Lafferty, McCallum, & Pereira, 2001), and Voted Perceptron (Collins, 2002) are widely used information extraction models.

Information extraction has been applied, for instance, to part-of-speech tagging (Ratnaparkhi, 1998), named entity recognition (Zhang, Pan, & Zhang, 2004) and table extraction (Pinto, McCallum, Wei, & Croft, 2003).

2.3. Search using title information

Title information is useful for document retrieval.

In the system Citeseer, for instance, Giles et al. managed to extract titles from research papers and make use of the extracted titles in *metadata* search of papers (Giles et al., 2003).

In web search, the title fields (i.e., file properties) and anchor texts of web pages (HTML documents) can be viewed as 'titles' of the pages (Evans, Klavans, & McKeown, 2004). Many search engines seem to utilize them for web page retrieval (Gheel & Anderson, 1999; Kobayashi & Takeda, 2000; Murphy, 1998; Yi & Sundaresan, 2000). Zhang and Dimitroff (2004), found that web pages with well-defined metadata are more easily retrieved than those without well-defined metadata.

To the best of our knowledge, no research has been conducted on using extracted titles from general documents (e.g., Office documents) for search of the documents.

3. Motivation and problem setting

We consider the issue of automatically extracting titles from general documents.

By general documents, we mean documents that belong to one of any number of specific genres. The documents can be presentations, books, book chapters, technical papers, brochures, reports, memos, specifications,

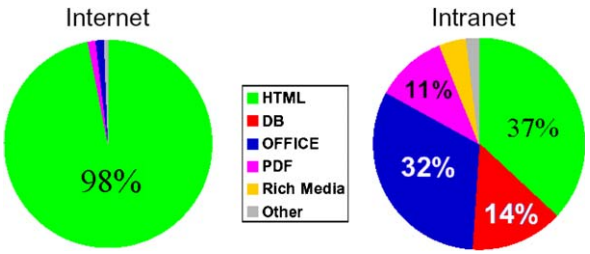

Fig. 1. Distributions of file formats in internet and intranet.

letters, announcements, or resumes. General documents are more widely available in digital libraries, intranets, and internet, and thus investigation on title extraction from them is sorely needed.

Fig. 1 shows an estimate on distributions of file formats on intranet and internet (Littlefield, 2002). Office and PDF are the main file formats on the intranet. Even on the internet, the documents in the formats are still not negligible, given its extremely large size. In this paper, without loss of generality, we take Office documents as an example.

For Office documents, users can define titles as file properties using a feature provided by Office. We found in an experiment, however, that users seldom use the feature and thus titles in file properties are usually very inaccurate. That is to say, titles in file properties are usually inconsistent with the 'true' titles in the file bodies that are created by the authors and are visible to readers. We collected 6000 Word and 6000 PowerPoint documents from an intranet and the internet and examined how many titles in the file properties are correct. We found that surprisingly the accuracy was only 0.265 (cf., Section 6.3 for details). A number of reasons can be considered. For example, if one creates a new file by copying an old file, then the file property of the new file will also be copied from the old file.

In another experiment, we found that Google uses the titles in file properties of Office documents in search and browsing, but the titles are not very accurate. We created 50 queries to search Word and PowerPoint documents and examined the top 15 results of each query returned by Google. We found that nearly all the titles presented in the search results were from the file properties of the documents. However, only 0.272 of them were correct.

Actually, 'true' titles usually exist at the beginnings of the bodies of documents. If we can accurately extract the titles from the bodies of documents, then we can exploit reliable title information in document processing. This is exactly the problem we address in this paper.

More specifically, given a Word document, we are to extract the title from the top region of the first page. Given a PowerPoint document, we are to extract the title from the first slide. A title sometimes consists of a main title and one or two subtitles. We only consider the extraction of the main title.

As baselines for title extraction, we use that of always using the first lines as titles and that of always using the lines with largest font sizes as titles.

Next, we define a 'specification' for human judgments in title data annotation. The annotated data will be used in training and testing of the title extraction methods.

Summary of the specification: The title of a document should be identified on the basis of common sense, if there is no difficulty in the identification. However, there are many cases in which the identification is not easy. There are some rules defined in the specification that guide identification for such cases. The rules include "a title is usually in consecutive lines in the same format", "a document can have no title", "titles in images are not considered", "a title should not contain words like 'draft', 'whitepaper', etc.", "if it is difficult to determine which is the title, select the one in the largest font size", and "if it is still difficult to determine which is the title, select the first candidate". (The specification covers all the cases we have encountered in data annotation.)

Figs. 2 and 3 show examples of Office documents from which we conduct title extraction. In Fig. 2, 'Differences in Win32 API Implementations among Windows Operating Systems' is the title of the Word document. 'Microsoft Windows' on the top of this page is a picture and thus is ignored. In Fig. 3, 'Building

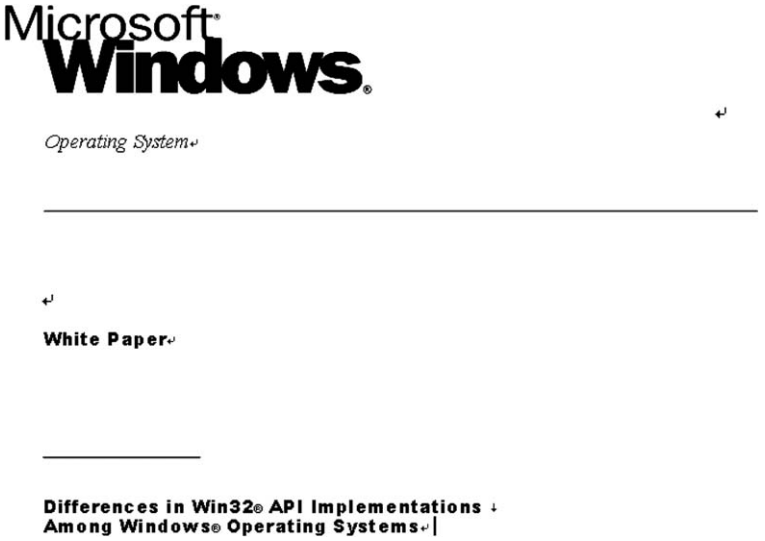

Fig. 2. Title extraction from Word document.

Fig. 3. Title extraction from PowerPoint document.

Competitive Advantages through an Agile Infrastructure' is the title of the PowerPoint document (note that the phrase "Agile Infrastructure" is in a different font style).

We have developed a tool for annotation of titles by human annotators. Fig. 4 shows a snapshot of the tool.

4. Title extraction method

4.1. Outline

Title extraction based on machine learning consists of training and extraction. The same pre-processing step occurs before training and extraction.

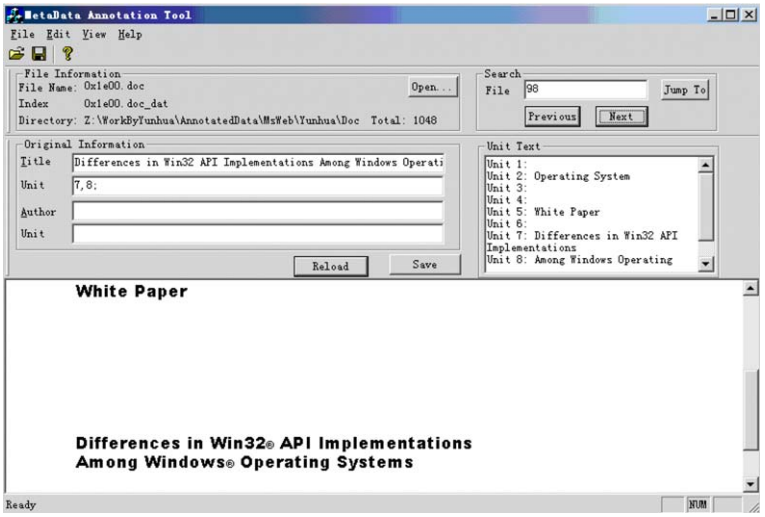

Fig. 4. Title annotation tool.

During pre-processing, from the top region of the first page of a Word document or the first slide of a PowerPoint document a number of units for processing are extracted. If a line (lines are separated by 'return' symbols) only has a single format, then the line will become a unit. If a line has several parts and each of them has its own format, then each part will become a unit. Each unit will be treated as an instance in learning. A unit contains not only content information (linguistic information) but also formatting information. The input to pre-processing is a document and the output of pre-processing is a sequence of units (instances). Fig. 5 shows the units obtained from the document in Fig. 2.

In learning, the input is sequences of units where each sequence corresponds to a document. We take labeled units (labeled as title_begin, title_end, or other) in the sequences as training data and construct models for identifying whether a unit is title_begin title_end, or other. We employ five types of models: Perceptron with Uneven Margins, Maximum Entropy (ME), Maximum Entropy Markov Model (MEMM), Voted Perceptron Model (VP), and Conditional Random Fields (CRF).

In extraction, the input is a sequence of units from one document. We employ one type of model to identify whether a unit is title_begin, title_end, or other. We then extract units from the unit labeled with 'title_begin' to the unit labeled with 'title_end'. The result is the extracted title of the document.

The unique characteristic of our approach is that we mainly utilize formatting information for title extraction. Our assumption is that although general documents vary in styles, their formats have certain patterns and we can learn and utilize the patterns for title extraction. This is in contrast to the work by Han et al., in which only linguistic features are used for extraction from research papers.

Unit 1:
Unit 2: [text="Operating System", alignment=left, boldface=false, largest_font_size=true, next_unit_is_empty=true, previous_unit_is_empty=true, ...]
Unit 3:
Unit 4:
Unit 5: [text="White Paper", alignment=left, boldface=true, largest_font_size=false, next_unit_is_empty=true, previous_unit_is_empty=true, ...]
Unit 6:
...

Fig. 5. Example of units.

4.2. Models

The five models actually can be considered in the same metadata extraction framework (see Fig. 6). That is why we apply them together to our current problem.

Each input is a sequence of instances $x_1x_2\cdots x_k$ together with a sequence of labels $y_1y_2\cdots y_k$. x_i and y_i represents an instance and its label, respectively ($i = 1,2,\ldots,k$). Recall that an instance here represents a unit. A label represents title_begin, title_end, or other. Here, k is the number of units in a document.

In learning, we train a model which can be generally denoted as a conditional probability distribution $P(Y_1\cdots Y_k|X_1\cdots X_k)$ where X_i and Y_i denote random variables taking instance x_i and label y_i as values, respectively ($i = 1,2,\ldots,k$).

We can make assumptions about the general model in order to make it simple enough for training.

For example, we can assume that $Y_1,\ldots,Y_k$ are independent of each other given $X_1,\ldots,X_k$. Thus, we have

$$P(Y_1\cdots Y_k|X_1\cdots X_k) = P(Y_1|X_1)\cdots P(Y_k|X_k)$$

In this way, we decompose the model into a number of classifiers. We train the classifiers locally using the labeled data. As the classifier, we employ the Perceptron or Maximum Entropy model.

In this paper, for Perceptron, we actually employ an improved variant of it, called Perceptron with Uneven Margin (Li, Zaragoza, Herbrich, Shawe-Taylor, & Kandola, 2002). This version of Perceptron can work well especially when the number of positive instances and the number of negative instances differ greatly, which is exactly the case in our problem.

We can also assume that the first order Markov property holds for $Y_1,\ldots,Y_k$ given $X_1,\ldots,X_k$. Thus, we have

$$P(Y_1\cdots Y_k|X_1\cdots X_k) = P(Y_1|X_1)\cdots P(Y_k|Y_{k-1}X_k)$$

Again, we obtain a number of classifiers. However, the classifiers are conditioned on the previous label. When we employ the Maximum Entropy model as a classifier, the models become a Maximum Entropy Markov Model. That is to say, this model is more precise.

If we do not make the above assumptions, we can make use of Conditional Random Fields (CRF), which is trained globally (i.e., at entire sequence level). CRF usually performs better than the other models. Voted Perceptron (VP) proposed by Collins (2002) can be viewed as a simplified version of CRF. We use both CRF and VP in this paper.

In extraction, given a new sequence of instances, we resort to one of the constructed models to assign a sequence of labels to the sequence of instances, i.e., perform extraction.

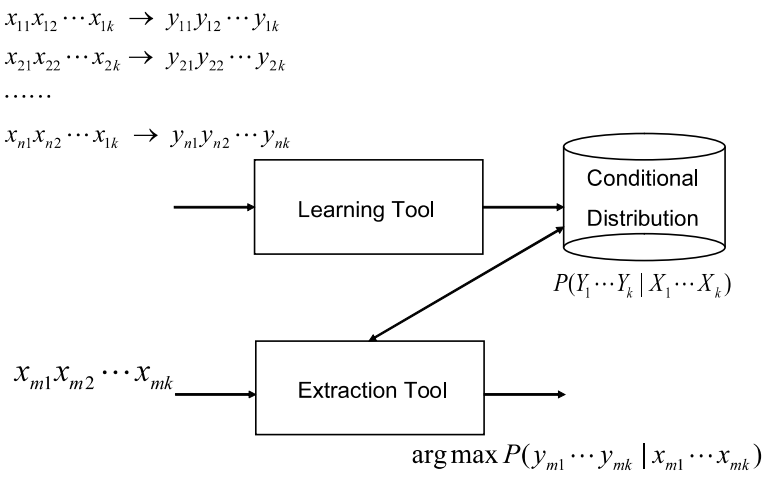

Fig. 6. Metadata extraction model.

For Perceptron and ME, we assign labels locally and combine the results globally later using heuristics. Specifically, we first identify the most likely title_begin. Then we find the most likely title_end within three units after the title_begin. Finally, we extract as a title the units between the title_begin and the title_end.

For MEMM, VP and CRF, we employ the Viterbi algorithm to find the globally optimal label sequence.

4.3. Features

There are two types of features: format features and linguistic features. We mainly use the former. The features are used for both the title-begin and the title-end classifiers.

4.3.1. Format features

Font size: There are four binary features that represent the normalized font size of the unit (recall that a unit has only one type of font).

If the font size of the unit is the largest in the document, then the first feature will be 1, otherwise 0. If the font size is the smallest in the document, then the fourth feature will be 1, otherwise 0. If the font size is above the average font size and not the largest in the document, then the second feature will be 1, otherwise 0. If the font size is below the average font size and not the smallest, the third feature will be 1, otherwise 0.

It is necessary to conduct normalization on font sizes. For example, in one document the largest font size might be '12 pt', while in another the smallest one might be '18 pt'.

Boldface: This binary feature represents whether or not the current unit is in boldface.

Alignment: There are four binary features that respectively represent the location of the current unit: 'left', 'center', 'right', and 'unknown alignment'.

The following format features with respect to 'context' play an important role in title extraction.

Empty neighboring unit: There are two binary features that represent, respectively, whether or not the previous unit and the current unit are blank lines.

Font size change: There are two binary features that represent, respectively, whether or not the font size of the previous unit and the font size of the next unit differ from that of the current unit.

Alignment change: There are two binary features that represent, respectively, whether or not the alignment of the previous unit and the alignment of the next unit differ from that of the current one.

Same paragraph: There are two binary features that represent, respectively, whether or not the previous unit and the next unit are in the same paragraph as the current unit.

4.3.2. Linguistic features

The linguistic features are based on key words.

Positive word: This binary feature represents whether or not the current unit begins with one of the positive words. The positive words include 'title:', 'subject:', 'subject line:' For example, in some documents the lines of titles and authors have the same formats. However, if lines begin with one of the positive words, then it is likely that they are title lines.

Negative word: This binary feature represents whether or not the current unit begins with one of the negative words. The negative words include 'To', 'By', 'created by', 'updated by', etc. This is because titles usually do not start with such words.

There are more negative words than positive words. The above linguistic features are language dependent.

Word count: A title should not be too long. We heuristically create four intervals: $[1,2]$, $[3,6]$, $[7,9]$ and $[9,\infty]$ and define one feature for each interval. If the number of words in a title falls into an interval, then the corresponding feature will be 1; otherwise 0.

Ending character: This feature represents whether the unit ends with ':', '-', or other special characters. A title usually does not end with such a character.

5. Document retrieval method

We describe our method of document retrieval using extracted titles.

Typically, in information retrieval a document is split into a number of fields including body, title, and anchor text. A ranking function in search can use different weights for different fields of the document. Also, titles are typically assigned high weights, indicating that they are important for document retrieval. As explained previously, our experiment has shown that a significant number of documents actually have incorrect titles in the file properties, and thus in addition of using them we use the extracted titles as one more field of the document. By doing this, we attempt to improve the overall precision.

In this paper, we employ a modification of BM25 that allows field weighting (Robertson, Zaragoza, & Taylor, 2004). As fields, we make use of body, title, extracted title and anchor. First, for each term in the query we count the term frequency in each field of the document; each field frequency is then weighted according to the corresponding weight parameter:

$$wtf_t = \sum_f w_f tf_{tf}$$

Similarly, we compute the document length as a weighted sum of lengths of each field. Average document length in the corpus becomes the average of all weighted document lengths.

$$wdl = \sum_f w_f dl_f$$

$$BM25F = \sum_t \frac{wtf_t(k_1+1)}{k_1\left((1-b)+b\frac{wdl}{avwdl}\right)+wtf} \times \log\left(\frac{N}{n}\right)$$

In our experiments we used $k_1 = 1.8$, $b = 0.75$. Weight for content was 1.0, title was 10.0, anchor was 10.0, and extracted title was 5.0.

6. Experimental results

6.1. Data sets and evaluation measures

We used three data sets in our experiments.

First, we downloaded and randomly selected 5000 Word documents and 5000 PowerPoint documents from an intranet of Microsoft. We call it MS hereafter.

Second, we downloaded and randomly selected 500 Word and 500 PowerPoint documents from the DotGov and DotCom domains on the internet, respectively.

Third, we downloaded and randomly selected 4000 Word and 4000 PowerPoint documents written in three other languages, including 4000 documents in Chinese, 2000 documents in Japanese and 2000 documents in German. We named them as Chinese, Japanese and German, respectively in the following.

Fig. 7 shows the distributions of the genres of the documents. We see that the documents are indeed 'general documents' as we define them.

We manually labeled the titles of all the documents, on the basis of our specification.

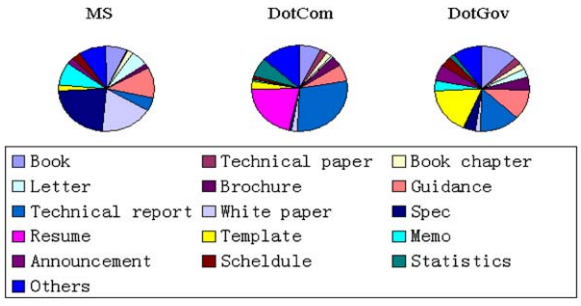

Fig. 7. Distributions of document genres.

Table 1
The portion of documents with titles

Type	Domain		
	MS (%)	DotCom (%)	DotGov (%)
Word	75.7	77.8	75.6
PowerPoint	82.1	93.4	96.4

Not all the documents have titles. Table 1 shows the percentages of the documents in the first two data sets having titles. We see that DotCom and DotGov have more PowerPoint documents with titles than MS. This might be because PowerPoint documents published on the internet are more formal than those on the intranet.

In our experiments, we conducted evaluations on title extraction in terms of precision, recall, and F-measure. The evaluation measures are defined as follows:

$$\text{Precision: } P = A/(A + B)$$
$$\text{Recall: } R = A/(A + C)$$
$$\text{F-measure: } F1 = 2PR/(P + R)$$

Here, A, B, C, and D are numbers of documents as those defined in Table 2.

6.2. Baselines

We test the accuracies of the two baselines described in Section 3. They are denoted as 'largest font size' and 'first line', respectively.

6.3. Accuracy of titles in file properties

We investigate how many titles in the file properties of the documents are reliable. We view the titles annotated by humans as true titles and test how many titles in the file properties can approximately match with the true titles. We use Edit Distance to conduct the approximate match. (approximate match is only used in this evaluation). This is because sometimes human annotated titles can be slightly different from the titles in file properties on the surface (e.g., contain extra spaces) (see Table 3).

Given string A and string B:

$$\text{if } ((D = 0) \text{ or } (D/(La + Lb) < \theta)) \text{ then string } A = \text{string } B$$

D: Edit distance between string A and string B
La: length of string A
Lb: length of string B
θ: 0.1

6.4. Comparison with baselines

We conducted title extraction from the first data set (Word and PowerPoint in MS). As the model, we used Perceptron.

Table 2
Contingence table with regard to title extraction

	Is title	Is not title
Extracted	A	B
Not extracted	C	D

Table 3
Accuracies of titles in file properties

File type	Domain	Precision	Recall	*F*1
Word	MS	0.299	0.311	0.305
	DotCom	0.210	0.214	0.212
	DotGov	0.182	0.177	0.180
PowerPoint	MS	0.229	0.245	0.237
	DotCom	0.185	0.186	0.186
	DotGov	0.180	0.182	0.181

Table 4
Accuracies of title extraction with Word

		Precision	Recall	*F*1
Model	Perceptron	0.810	0.837	0.823
Baselines	Largest font size	0.700	0.758	0.727
	First line	0.707	0.767	0.736

Table 5
Accuracies of title extraction with PowerPoint

		Precision	Recall	*F*1
Model	Perceptron	0.875	0. 895	0.885
Baselines	Largest font size	0.844	0.887	0.865
	First line	0.639	0.671	0.655

We conduct fourfold cross validation. Thus, all the results reported here are those averaged over four trials. Tables 4 and 5 show the results. We see that Perceptron significantly outperforms the baselines. In the evaluation, we use exact matching between the true titles annotated by humans and the extracted titles.

We see that the machine learning approach can achieve good performance in title extraction. For Word documents both precision and recall of the approach are 8 percent higher than those of the baselines. For PowerPoint both precision and recall of the approach are 2 percent higher than those of the baselines.

We conduct significance tests. The results are shown in Table 6. Here, 'Largest' denotes the baseline of using the largest font size, 'First' denotes the baseline of using the first line. The results indicate that the improvements of machine learning over baselines are statistically significant (in the sense p-value < 0.05).

We see, from the results, that the two baselines can work well for title extraction, suggesting that font size and position information are most useful features for title extraction. However, it is also obvious that using only these two features is not enough. There are cases in which all the lines have the same font size (i.e., the largest font size), or cases in which the lines with the largest font size only contain general descriptions like 'Confidential', 'White paper', etc. For those cases, the 'largest font size' method cannot work well. For similar reasons, the 'first line' method alone cannot work well, either. With the combination of different features (evidence in title judgment), Perceptron can outperform Largest and First.

Table 6
Sign test results

Documents type	Sign test between	*p*-value
Word	Perceptron vs. Largest	3.59e−26
	Perceptron vs. First	7.12e−10
PowerPoint	Perceptron vs. Largest	0.010
	Perceptron vs. First	5.13e−40

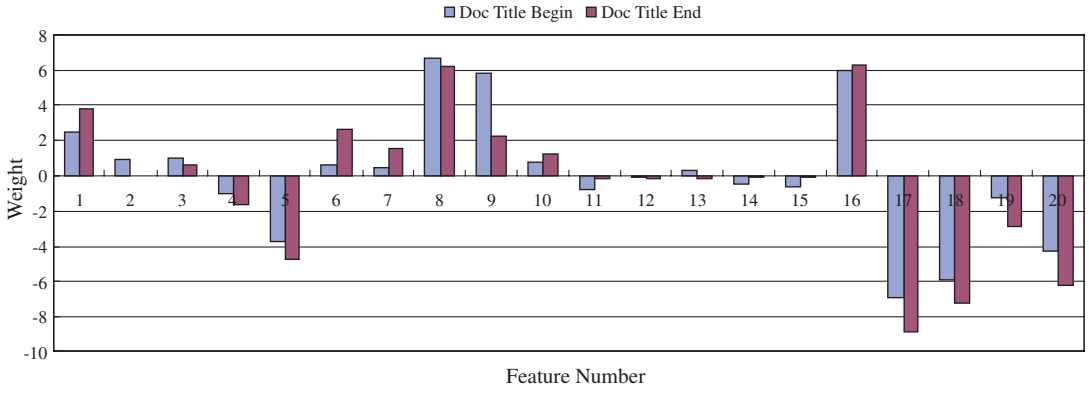

Fig. 8. Weights of some features in Perceptron models for Word.

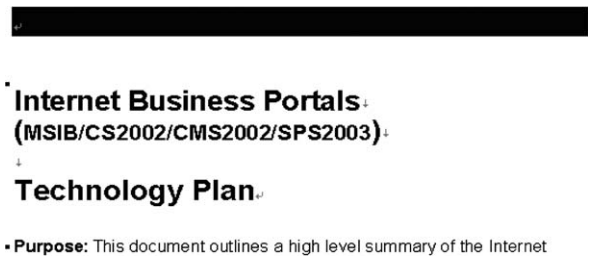

Fig. 9. An example Word document.

We investigate the performance of solely using linguistic features. We found that it does not work well. It seems that the format features play important roles and the linguistic features are supplements. Fig. 8 shows the weights of some randomly selected features in the Perceptron models. Features 1–10 are format features and features 11–20 are linguistic features. Although linguistic features tend to have larger weights, usually titles do not contain such features.

We conducted an error analysis on the results of Perceptron. We found that the errors fell into three categories. (1) About one-third of the errors were related to 'hard cases'. In these documents, the layouts of the first pages were difficult to understand, even for humans. Fig. 9 and 10 shows examples. (2) Nearly one-fourth

Fig. 10. An example PowerPoint document.

of the errors were from the documents which do not have true titles but only contain bullets. Since we conduct extraction from the top regions, it is difficult to get rid of these errors with the current approach. (3) Confusions between main titles and subtitles were another type of error. Since we only labeled the main titles as titles, the extractions of both titles were considered incorrect. This type of error does little harm to document processing like search, however.

6.5. Comparison between models

To compare the performance of different machine learning models, we conducted another experiment. Again, we perform fourfold cross validation on the first data set (MS). Tables 7 and 8 show the results of all the five models.

It turns out that CRF performs the best, followed by Perceptron and VP, then MEMM, and ME performs the worst. In general, the Markovian models perform better than or as well as their classifier counterparts. This seems to be because the Markovian models are based on local histories and thus can more effectively make use of context information than the classifier models. The Perceptron based models perform better than the ME based counterparts. This seems to be because the Perceptron based models are created to make better classifications, while ME models are constructed for better prediction. It is not surprising to see that CRF performs the best, because it employs a globally (sequence level) trained model. Because our main focus in this paper is to conduct feasibility study of title extraction from general documents, we utilized the simplest model Perceptron in our experiments.

6.6. Domain adaptation

We apply the model trained with the first data set (MS) to the second data set (DotCom and DotGov). Tables 9–12 show the results.

Table 7
Comparison between different learning models for title extraction with Word

Model	Precision	Recall	*F*1
Perceptron	0.810	0.837	0.823
ME	0.801	0.621	0.699
MEMM	0.797	0.824	0.810
VP	0.827	0.823	0.825
CRF	0.834	0.843	0.838

Table 8
Comparison between different learning models for title extraction with PowerPoint

Model	Precision	Recall	*F*1
Perceptron	0.875	0.895	0.885
ME	0.753	0.766	0.759
MEMM	0.841	0.861	0.851
VP	0.873	0.896	0.885
CRF	0.880	0.891	0.886

Table 9
Accuracies of title extraction with Word in DotGov

		Precision	Recall	*F*1
Model	Perceptron	0.716	0.759	0.737
Baselines	Largest font size	0.549	0.619	0.582
	First line	0.462	0.521	0.490

Table 10
Accuracies of title extraction with PowerPoint in DotGov

		Precision	Recall	*F*1
Model	Perceptron	0.900	0.906	0.903
Baselines	Largest font size	0.871	0.888	0.879
	First line	0.554	0.564	0.559

Table 11
Accuracies of title extraction with Word in DotCom

		Precision	Recall	*F*1
Model	Perceptron	0.832	0.880	0.855
Baselines	Largest font size	0.676	0.753	0.712
	First line	0.577	0.643	0.608

Table 12
Performance of PowerPoint document title extraction in DotCom

		Precision	Recall	*F*1
Model	Perceptron	0.910	0.903	0.907
Baselines	Largest font size	0.864	0.886	0.875
	First line	0.570	0.585	0.577

From the results, we see that the models can be adapted to different domains well. There is almost no drop in accuracy. The results indicate that the patterns of title formats exist across different domains, and it is possible to construct a domain independent model by mainly using formatting information.

6.7. Language adaptation

We apply the model trained with the data in English (MS) to the data set in third data set (Chinese, Japanese and German). Tables 13–18 show the results.

We see that the models can be adapted to a different language. There are only small drops in accuracy. Obviously, the linguistic features do not work for other languages, but the effect of not using them is negligible. The results indicate that the patterns of title formats exist across different languages.

Table 13
Accuracies of title extraction with Word in Chinese

		Precision	Recall	*F*1
Model	Perceptron	0.861	0.843	0.852
Baselines	Largest font size	0.436	0.908	0.589
	First line	0.689	0.604	0.644

Table 14
Accuracies of title extraction with PowerPoint in Chinese

		Precision	Recall	*F*1
Model	Perceptron	0.858	0.878	0.868
Baselines	Largest font size	0.823	0.950	0.882
	First line	0.845	0.669	0.747

Table 15
Accuracies of title extraction with Word in Japanese

		Precision	Recall	*F*1
Model	Perceptron	0.744	0.741	0.743
Baselines	Largest font size	0.436	0.908	0.589
	First line	0.689	0.604	0.644

Table 16
Accuracies of title extraction with PowerPoint in Japanese

		Precision	Recall	*F*1
Model	Perceptron	0.875	0.910	0.892
Baselines	Largest font size	0.828	0.965	0.891
	First line	0.803	0.616	0.697

Table 17
Accuracies of title extraction with Word in German

		Precision	Recall	*F*1
Model	Perceptron	0.723	0.729	0.726
Baselines	Largest font size	0.436	0.908	0.589
	First line	0.689	0.604	0.644

Table 18
Accuracies of title extraction with PowerPoint in German

		Precision	Recall	*F*1
Model	Perceptron	0.820	0.841	0.830
Baselines	Largest font size	0.729	0.898	0.805
	First line	0.797	0.586	0.675

From the domain adaptation and language adaptation results, we conclude that the use of formatting information is the key to a successful extraction from general documents.

6.8. Search with extracted titles

We performed experiments on using title extraction for document retrieval. As a baseline, we employed BM25 without using extracted titles. The ranking mechanism was as described in Section 5. The weights were determined manually. We did not conduct optimization on the weights.

The evaluation was conducted on a corpus of 1.3 M documents crawled from the intranet of Microsoft using 100 evaluation queries obtained from this intranet's search engine query logs. Fifty queries were from the most popular set, while another 50 queries were chosen randomly. Users were asked to provide judgments of the degree of document relevance from a scale of 1–5 (1 – means detrimental, 2 – bad, 3 – fair, 4 – good, and 5 – excellent).

Fig. 11[2] shows the results. In the chart two sets of precision results were obtained by either considering good or excellent documents as relevant (left three bars with relevance threshold 0.5), or by considering only excellent documents as relevant (right three bars with relevance threshold 1.0).

[2] For colour interpretation the reader is referred to see the web version of this figure.

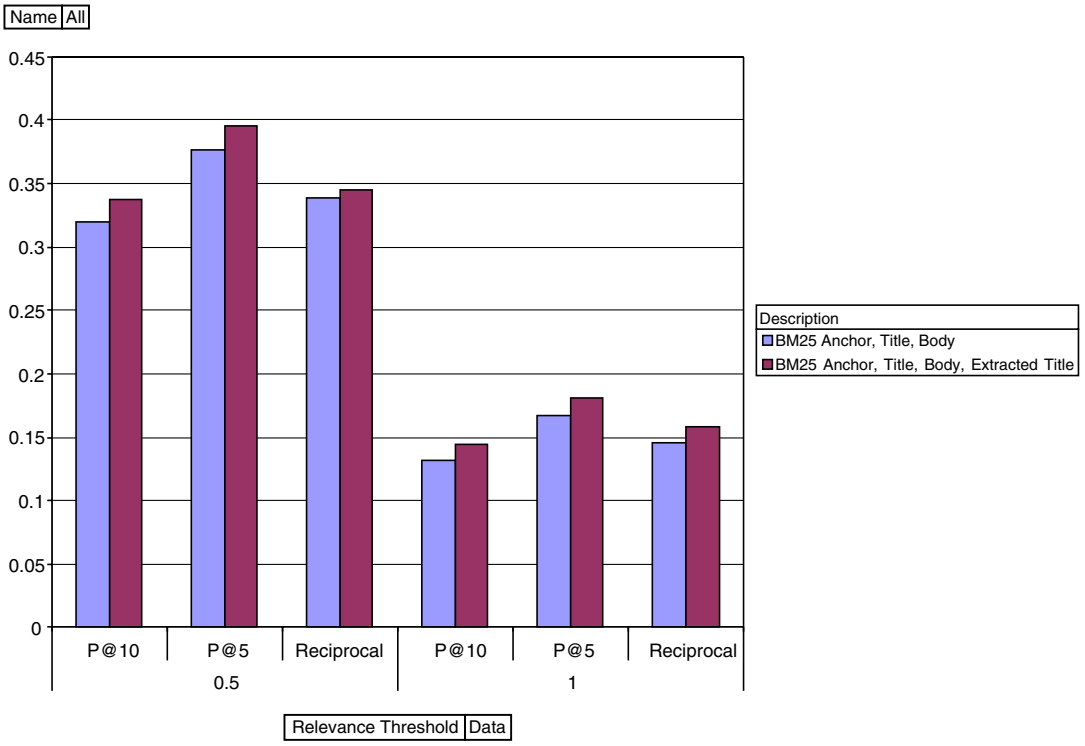

Fig. 11. Search ranking results.

Fig. 11 shows different document retrieval results with different ranking functions in terms of precision @10, precision @5 and reciprocal rank:

- Blue bar – BM25 including the fields body, title (file property), and anchor text.
- Purple bar – BM25 including the fields body, title (file property), anchor text, and *extracted title*.

With the additional field of extracted title included in BM25 the precision @10 increased from 0.132 to 0.145, or by ~10%. Thus, it is safe to say that the use of extracted title can indeed improve the precision of document retrieval.

7. Conclusion

In this paper, we have investigated the problem of automatically extracting titles from general documents. We have tried using a machine learning approach to address the problem.

Previous work showed that the machine learning approach can work well for metadata extraction from research papers. In this paper, we showed that the approach can work for extraction from general documents as well. Our experimental results indicated that the machine learning approach can work significantly better than the baselines in title extraction from Office documents. Previous work on metadata extraction mainly used linguistic features in documents, while we mainly used formatting information. It appeared that using formatting information is a key for successfully conducting title extraction from general documents.

We tried different machine learning models including Perceptron, Maximum Entropy, Maximum Entropy Markov Model, Voted Perceptron and Conditional Random Fields. We found that the performance of the Conditional Random Fields models were the best. We applied models constructed in one domain to another domain and applied models trained in one language to another language. We found that the accuracies did not drop substantially across different domains and across different languages, indicating that the

models were generic. We also attempted to use the extracted titles in document retrieval. We observed a significant improvement in document ranking performance for search when using extracted title information. All the above investigations were not conducted in previous work, and through our investigations we verified the generality and the significance of the title extraction approach.

Acknowledgements

We thank Chunyu Wei and Bojuan Zhao for their work on data annotation. We acknowledge Jinzhu Li for his assistance in conducting the experiments. We thank Ming Zhou, John Chen, and Jun Xu for their valuable comments on early versions of this paper. We also thank the anonymous reviewers of this paper for their making many valuable comments.

References

Berger, A. L., Della Pietra, S. A., & Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics, 22*, 39–71.

Crystal, A., & Land, P. (2003). Metadata and Search Global Corporate Circle DCMI 2003 Workshop. Available from http://dublincore.org/groups/corporate/Seattle/.

Collins, M. (2002). Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of conference on empirical methods in natural language processing* (pp. 1–8).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Chieu, H. L., & Ng, H. T. (2002). A maximum entropy approach to information extraction from semi-structured and free text. In *Proceedings of the eighteenth national conference on artificial intelligence* (pp. 768–791).

Evans, D. K., Klavans, J. L., & McKeown, K. R. (2004). Columbia newsblaster: multilingual news summarization on the Web. In *Proceedings of human language technology conference/North American chapter of the association for computational linguistics annual meeting* (pp. 1–4).

Ghahramani, Z., & Jordan, M. I. (1997). Factorial hidden markov models. *Machine Learning, 29*, 245–273.

Gheel, J., & Anderson, T. (1999). Data and metadata for finding and reminding. In *Proceedings of the 1999 international conference on information visualization* (pp. 446–451).

Giles, C. L., Petinot, Y., Teregowda, P. B., Han, H., Lawrence, S., & Rangaswamy, A., et al. (2003). eBizSearch: a niche search engine for e-Business. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 413–414).

Giuffrida, G., Shek, E. C., & Yang, J. (2000). Knowledge-based metadata extraction from PostScript files. In *Proceedings of the fifth ACM conference on digital libraries* (pp. 77–84).

Han, H., Giles, C. L., Manavoglu, E., Zha, H., Zhang, Z., & Fox, E. A. (2003). Automatic document metadata extraction using support vector machines. In *Proceedings of the third ACM/IEEE-CS joint conference on digital libraries* (pp. 37–48).

Kobayashi, M., & Takeda, K. (2000). Information retrieval on the Web. *ACM Computing Surveys, 32*, 144–173.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning* (pp. 282–289).

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., & Kandola, J. S., (2002). The perceptron algorithm with uneven margins. In *Proceedings of the nineteenth international conference on machine learning* (pp. 379–386).

Liddy, E. D., Sutton, S., Allen, E., Harwell, S., Corieri, S., & Yilmazel, O., et al. (2002). Automatic metadata generation & evaluation. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 401–402).

Littlefield, A. (2002). Effective enterprise information retrieval across new content formats. In *Proceedings of the seventh search engine conference*. Available from http://www.infonortics.com/searchengines/sh02/02prog.html.

Mao, S., Kim, J. W., & Thoma, G. R. (2004). A dynamic feature generation system for automated metadata extraction in preservation of digital materials. In *Proceedings of the first international workshop on document image analysis for libraries* (pp. 225–232).

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the seventeenth international conference on machine learning* (pp. 591–598).

Murphy, L. D. (1998). Digital document metadata in organizations: roles, analytical approaches, and future research directions. In *Proceedings of the thirty-first annual Hawaii international conference on system sciences* (pp. 267–276).

Peng, F., & McCallum, A. (2004). Accurate information extraction from research papers using conditional random fields. In *Proceedings of the human language technology conference/North American chapter of the association for computational linguistics annual meeting* (pp. 329–336).

Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 235–242).

Ratnaparkhi, A. (1998). Unsupervised statistical models for prepositional phrase attachment. In *Proceedings of the seventeenth international conference on computational linguistics* (pp. 1079–1085).

Robertson, S., Zaragoza, H., & Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of ACM thirteenth conference on information and knowledge management* (pp. 42–49).

Yi, J., & Sundaresan, N. (2000). Metadata based Web mining for relevance. In *Proceedings of the 2000 international symposium on database engineering & applications* (pp. 113–121).

Yilmazel, O., Finneran, C. M., & Liddy, E. D. (2004). MetaExtract: an NLP system to automatically assign metadata. In *Proceedings of the 2004 joint ACM/IEEE conference on digital libraries* (pp. 241–242).

Zhang, J., & Dimitroff, A. (2004). Internet search engines' response to metadata Dublin Core implementation. *Journal of Information Science, 30*, 310–320.

Zhang, L., Pan, Y., & Zhang, T. (2004). Recognising and using named entities: focused named entity recognition using machine learning. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 281–288).