Robust Template Identification of Scanned Documents

Xiaofan Feng[1], Abdou Youssef[1] and Sithu D. Sudarsan[2]
[1]*Department of Computer Science, The George Washington University, Washington DC, U.S.A.*
[2]*Office of Science and Engineering Labs, Center for Devices and Radiological Health, US Food and Drug Administration, Silver Spring, MD, U.S.A.*

Keywords: Scanned Document Identification, Maximum A-Posterior Estimation, Information Retrieval.

Abstract: Identification of low-quality scanned documents is not trivial in real-world settings. Existing research mainly focusing on similarity-based approaches rely on perfect string data from a document. Also, studies using image processing techniques for document identification rely on clean data and large differences among templates. Both these approaches fail to maintain accuracy in the context of noisy data or when document templates are too similar to each other. In this paper, a probabilistic approach is proposed to identify the document template of scanned documents. The proposed algorithm works on imperfect OCR output and document collections containing very similar templates. Through experiment and analysis, this novel probabilistic approach is shown to achieve high accuracy on different data sets.

1 INTRODUCTION

Although electronic documents have become prevalent, governments and enterprises still possess a large volume of paper documents. A major task is to digitize, label, and extract information from these paper documents. Many documents used in enterprises and governments are typically derived from templates, especially forms completed by users, e.g., tax forms, medical forms, job application forms, etc. Given a set of templates and a scanned paper document, an open problem is to quickly and accurately identify which template this scanned document was originally derived from (Esser et al., 2011). To solve this problem, a number of systems based on labeled information have been proposed and developed (Cunningham et al., 2002; T. S. Jayram et al., 2006) The labeled information is usually manually generated, making document identification a time-consuming and expensive process. Studies have been performed to use image features to match a scanned document to its template (Hu et al., 2000). Some of these studies still require labeled information (Esser et al., 2011), while others require consistent high-quality data in order to function properly.

Identifying documents in a repository of scanned documents via manual labeling is somewhat inefficient and expensive. Most automatic image processing techniques require clean data making these techniques ineffective in the presence of noise. Another drawback of those techniques is that they cannot correctly correlate a document to a template when the templates are too similar to each other, which is a serious problem because many form templates in governments and enterprises are near identical. As such, a robust system to automatically identify the originating template of a scanned document is necessary. Once the template has been correctly identified, existing techniques can be utilized to retrieve specific information.

Now, suppose that a scanned document has been successfully matched to its template. To extract information from the scanned document, one of the first steps commonly applied is Optical Character Recognition (OCR). Even though state-of-the-art OCR techniques are highly accurate in recognizing printed words and characters in clean simple-formatted documents, these techniques still are error-prone when the input is noisy or the document-format is complicated. The accuracy of an OCR engine becomes less reliable for documents that have become distorted under scanning, aging, or folding. Furthermore, OCR output tends to be inaccurate when the document contains multiple columns and text of different font types and font sizes.

This paper introduces an efficient method to identify a document to its originating template by utilizing the results generated by OCR. First, the text contained in the templates is retrieved. Then a scanned document image is OCRed. This result is compared

Feng X., Youssef A. and Sudarsan S..
Robust Template Identification of Scanned Documents.
DOI: 10.5220/0004144601030110
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 103-110
ISBN: 978-989-8565-29-7

to all the templates' text using a probabilistic model, and the most likely template is selected. The proposed approach works well on noisy documents and does not require manual intervention or labeling. It also works well on both document templates which are significantly different from each other and on document templates which are nearly identical to each other. The approach can be easily incorporated into other OCR techniques to reduce document processing time.

2 RELATED WORKS

Document template identification has been studied extensively. In this section we summarize the related work. A survey of the literature quickly reveals that existing approaches fall into two categories: Information Retrieval approaches, and Image Feature based approaches.

2.1 Information Retrieval Approaches

To identify the template of a scanned document with information retrieval techniques, the textual information of the document is treated as a query against a database of template documents (Salton, 1986). The template document which has the highest rank is taken as the identification result.

Cosine similarity is a common similarity measurement used in document query or identification. (Press et al., 2007; Tan et al., 2005; Salton et al., 1975). Given two documents represented as term-frequency vectors v_1 and v_2, their cosine similarity is defined as:

$$similarity\{v_1, v_2\} = \cos(\theta) = \frac{v_1 v_2}{\|v_1\|\|v_2\|} \quad (1)$$

These term-frequency vectors can be applied with local and global functions to improve the results in different scenarios.

Singular Value Decomposition (SVD) is another well developed technique that is widely used in document query or identification. There are variations of this technique, namely latent semantic indexing (LSI) (Deerwester,), probabilistic latent semantic indexing (pLSI) (Hofmann, 1999), and latent dirichlet allocation (LDA) (Blei et al., 2003). SVD based techniques sometimes achieve better results than just using terms for retrieval. Taking a lower rank of SVD can help filter out the noise in the document-term matrix. OCR may recognize some terms incorrectly. These terms tend to appear together within a set of same terms for the documents generated from the same template. SVD can be explored as the fill-in data that do not frequently occur can be filtered out as noise with SVD.

2.2 Image Feature Approaches

Image-feature based approaches have also been extensively studied for document template identification. To identify the template of a certain document, different features of the image have been selected, (Lu and Tan, 2004; Hu et al., 2000; Zheng et al., 2005; Jinhui Liu, 2000; Shin et al., 2001). Based on different features in use, different similarity measurements have been proposed. In (Jinhui Liu, 2000) and (Zheng et al., 2005), the image is classified with geometric line information. These methods require image data to be clean and noise-free, and do not work on forms containing free-size cells. Free-size cells are those cells whose sizes will adjust with the filled-in data. Image based approaches cannot work on forms generated with the same form structure but having different contents. The ZoneSeg method used by (Esser et al., 2011) is chosen as representative of image feature approach to compare against our approach. In (Esser et al., 2011), this method has been claimed to have over 98% accuracy.

3 PROBLEM FORMULATION

In this section, we formulate the problem of scanned document identification, describe the intuitive approaches, and identify shortcomings in those approaches.

3.1 Problem Definition

A modified bag-of-words representation is used for the document. Each document is treated as a set of strings. This modified bag-of-words representation maintains the frequency of each string occurrence and does not disregard grammar. For example, unlike the traditional bag-of-words representation, this modified representation does not use stemming and does not disregard punctuation.

Let N_T denote the size of the document-templates set. Each template is denoted as T_i, with $i \in [1, N_T]$. Each template T_i can be represented as a set $\overline{T}_i$:

$$\overline{T}_i = \{s_1^i, s_2^i, \ldots, s_m^i\}$$

where $s_1^i, s_2^i, \ldots, s_m^i$ are the strings that appear in the template T_i. If a string appears multiple times in T_i, it is represented once in $\overline{T}_i$, and its frequency is recorded. The frequency (or count) of appearance for the strings in template T_i is denoted by

$$C_i = \{c_1^i, c_2^i, \ldots, c_m^i\}$$

Similarly, an unlabeled query document E, i.e., the document needing to be identified, is represented as a set $\overline{E}$:

$$\overline{E} = \{e_1, e_2, \ldots, e_n\}$$

where the e_j are the strings that appear in E, and the frequencies of these strings in E are

$$C_E = \{c_1^e, c_2^e, \ldots, c_n^e\}$$

The goal is to find the template T_k that the input document E was derived from, where $T_k \in \{T_1, T_2, \ldots, T_{N_T}\}$.

Lemma 3.1. *Suppose the OCR result of E is error-free. In this case,* $\overline{T}_k \subseteq \overline{E}$.

We have the case that $\overline{T}_k \subset \overline{E}$ when the input document E was filled-out by a user or annotated by a fax machine (e.g., adding the timestamp and the fax number). Since E was derived from the template T_k, all the strings in T_k will also be in E given that the OCR result is error-free. We can also have the case that $E = T_k$ in the event the input document E was just T_k scanned in, without any additional strings added by a user or a fax machine.

In this particular situation, common information retrieval techniques (e.g., cosine similarity) can produce decent results. However, many scanned and faxed documents cannot be OCRed error-free.

Observation 3.1. *Suppose E is derived from* T_k. *It may be the case that* $\overline{E} \cap \overline{T}_{j \neq k} \neq \emptyset$. *Let* $\overline{E} = A \cup B$, $A \subseteq \overline{T}_k$, $B \cap \overline{T}_k = \emptyset$. *One or more of the following cases may hold true:*

1. $A \cap \overline{T}_{j \neq k} \neq \emptyset$
2. $B \cap \overline{T}_{j \neq k} \neq \emptyset$
3. $(A \cup B) \cap \overline{T}_{j \neq k} \neq \emptyset$

The first case stems from the fact that the strings which appear in input E may also appear in templates other than T_k (due to the overlap of strings between templates). The second case stems from strings appearing in E and not appearing in T_k but appearing in other templates in the set, as a result of filled-in data. The third case can occur when both the first and second cases occur simultaneously.

The challenges of this problem are due to the imperfect output from OCR and the user fill-in data. Our goal is to be able to find the correct template T_k for E not only when E does not include all strings of T_k, but also when E contains information appearing on other templates $T_{j \neq k}, j \in [1, N_t]$, given that templates themselves can be very similar to each other.

Note that a document represented with the traditional bag-of-words model is not desired in this situation. Stemming removes grammatical information, which is useful in our setting. For example, a term appearing on a template in *singular* form should be treated differently from a term appearing in *plural* form. Comparing documents represented as a sequence of strings or as a sequence of words involves searching a space that is exponentially larger than our representation. Furthermore, for the current problem context, representation considering the order of the words (e.g., n-grams) is unsuitable for comparing documents since the OCR output will have errors, and word-order is not guaranteed in complicated document formats.

3.2 Intuitive Approaches

Given the problem formulation, there are a few intuitive solutions, which we discuss in this section.

The very direct and intuitive solution for this problem is as follows:

Given N_T different templates, if every template T_i has a subset of strings unique to T_i, i.e., occurring in T_i and in no other template, then these subsets of strings can be used as signatures. We denote the signature of template T_i with S_i. Since $\forall s \in S_i$, $s \in T_i$ and $s \notin S_j$, $\forall j \neq i$, if any of the strings of the signature can be found in the OCR result of E, we can identify the template. This approach is problematic, however.

Depending on the similarities among the templates in the set, the number $|S_i|$ of uniquely appearing strings differs across templates. For some template sets, we will be able to find many strings that uniquely identify each template. But for other template sets, we might have degenerate cases with no strings to uniquely identify some of the templates. In addition, as the number of templates grows in the set, the size of the signatures tend to become smaller. More importantly, the success of this approach relies on perfect OCR, with all strings from a query document E being extracted correctly. If all the strings in the signature S_i have been incorrectly recognized, it would not be possible to find the originating template. Furthermore, realistically speaking, there is no control over the information that will be filled-in in the query document E. Thus, we could have the case where E contains strings that are a part of signatures of multiple templates.

Another intuitive solution would be to use cosine-similarity. If E was derived from the template T_k, its vector should be close to T_k's vector. However, we have two caveats here: First, with OCR errors and user filled-in data, the vector representation of E might be dramatically different from T_k, resulting in a large distance between E and T_k. This may result in a false match when E's vector is closer to another template's vector. Second, if two templates T_k and T_j are

very similar (i.e., the vector space distance between T_k and T_j is very small), we will have the case where we are not able to determine whether E was derived from T_k or T_j.

4 A PROBABILISTIC METHOD FOR DOCUMENT IDENTIFICATION

In this section, we will discuss how to solve the document identification using a probabilistic approach in detail.

The basic idea of our probabilistic approach is to explore the difference in the feature space of similar templates. To identify the template of an unlabeled document E, both the similarity and the dissimilarity between E and a template T_i should be considered. The similarity serves as a filter to exclude the templates which E cannot belong to, while the dissimilarity is used to enlarge the differences between similar templates.

4.1 The Probabilistic Method

For a query document E, we first use OCR to extract all the strings and build the set C_E. If E was derived from the template T_k, there may be instances where $\overline{E} \cap \overline{T}_k \neq \emptyset$ and $\overline{E} \cap \overline{T}_j \neq \emptyset$, for some j and k where $j \neq k$. These instances result from two situations (or a combination of these situations, as defined in Observation 3.1). First, the templates may share common strings among themselves, that is, $\overline{T}_k \cap \overline{T}_j \neq \emptyset$, $k \neq j$. Second, the input may have been filled with information that was correctly or incorrectly recognized as a string occurring in another template T_j, $(\overline{E} \setminus \overline{T}_k) \cap \overline{T}_j \neq \emptyset$. For a string s appearing in input E and template T_k, it is possible that E was derived from T_k. It is also possible that E was derived from another template T_j, but the filled-in data in E contained s. To handle this uncertainty, we will use maximum a posteriori (MAP) estimation to estimate the most likely template from which E was derived.

From the set of templates $\{T_1, T_2, \ldots, T_{N_T}\}$, a union of the set of strings can be obtained, as $U_T = \bigcup_{i=1}^{N_T} \overline{T}_i$. Any string that is in E but not in U_T is ignored. Identifying which template T_i, $i \in [1, N_T]$, the input E was generated from is equivalent to deciding which template that the $E' = \overline{E} \cap U_T$ is most likely to follow.

For any string $e_i \in E'$, $e_i \in U_T$, the number of times e_i appears in E' is c_i^e. Suppose e_i appears in template T_i, with its count of occurrence in T_i being c_j^i. The case where $c_j^i \geq c_i^e$ may be a result of OCR errors with e_i being incorrectly recognized as other string(s). The case could also be that some of the e_i in E have been incorrectly identified, as well as some fill-in data containing e_i have been correctly recognized, but the total number of occurrence is still less than c_j^i. In the case of $c_j^i < c_i^e$, it is possible that all the strings e_i from template T_i have been correctly recognized by OCR, and fill-in data containing exact $c_i^e - c_j^i$ of e_i, it is also possible that not all c_j^i occurrences of e_i are correctly recognized, but the fill-in data containing more than $c_i^e - c_j^i$ occurrences of e_i are recognized.

To handle these uncertain situations, two probabilities are defined. In general, we can state that the OCR success in recognizing a string e_i follows a Bernoulli distribution with probability p. For the template T_i with e_i appearing c_j^i times, while e_i appears in E for $c_i^e \leq c_j^i$, the probability follows a Binomial distribution (c_j^i, p). A string e_i which appears in U_T follows a Bernoulli distribution with a probability of q that it actually comes from the fill-in data.

Given the input E as an observation and a set of templates $T_1, T_2, \ldots, T_{N_T}$, the template that maximizes the posterior probability will be selected as the template that E was derived from. The posterior probability is computed as:

$$\begin{aligned} \hat{i}_{MAP}(C_E) &= \arg\max_i p(T_i|C_E) \\ &= \arg\max_i \frac{p(C_E|T_i)p(T_i)}{p(C_E)} \\ &\propto \arg\max_i p(C_E|T_i)p(T_i) \end{aligned} \tag{2}$$

where $p(T_i)$ is the prior probability distribution of the template T_i. Based on the particular application, if knowledge regarding which template is more frequent is known, this probability can be assigned. In the case that there is no such prior knowledge, $p(T_i)$ can be assigned with the Uniform distribution, with $p(T_i) = 1/N_T$. This results in the MAP estimator being the Maximum likelihood estimator (MLE).

Now, we discuss how to compute $p(C_E|T_i)$. The recognition result of each string in the template T_i from which E has been derived can be taken as a random variable following a Binomial distribution of (c_j^i, p_i). The recognition results of all strings in template T_i are assumed to be independent identically distributed. The probability of having any string from $U_T \setminus T_i$, which is the probability of filled-in data is the same as strings occurring in other templates is q. Then the probability $p(C_E|T_i)$ can be precisely computed as the combination of these two distributions, which is too complicated. Under the assumptions $p >> q$, $p(c_i^e|T_i)$ can be simplified as: if $c_i^e \leq c_j^i$, all strings

are considered as being from the template; if $c_i^e \geq c_j^i$, c_j^i strings are assumed to be from the template, while $c_i^e - c_j^i$ are strings accidentally falling in U_T.

$$p(c_i^e|T_i) = \begin{cases} \binom{c_j^i}{c_i^e} p_i^{c_i^e} (1-p_i)^{c_j^i - c_i^e} & \text{if } c_i^e \leq c_j^i \\ p_i^{c_j^i} q^{c_i^e - c_j^i} & \text{if } c_i^e \geq c_j^i \end{cases} \quad (3)$$

Hence, given a template T_i, the conditional probability of having an observed data as E with its frequency vector as C_E can be computed approximately as:

$$p(C_E|T_i) = \prod_{l=1}^{n} p(c_l^e|T_i) \quad (4)$$

With Equation 3 and 4, the MAP estimator of template T_i can be computed. We use the logarithm of the probability as

$$\hat{i}_{MAP}(E) \propto \arg\max_i \log p(C_E|T_i)p(T_i) \quad (5)$$

This can be computed faster and avoid the underflow problems.

4.2 Considerations

The probabilistic model allows for situations when an important string is not properly recognized by OCR. At the same time, this model leverages the length of the words. The p_i used in our experiment is obtained by first OCRing the known templates and comparing with the ground-truth on each template and their OCRing output to obtain an estimation of p_i. It can be argued that this p_i does not reflect any particular error rate on different input E. However, it is useful to express the relative error rates among the templates.

5 EXPERIMENTAL PERFORMANCE EVALUATION

In this section, the proposed algorithm is evaluated on two data sets. Some properties of the data sets are discussed first, followed by a brief introduction to other algorithms we compare against. Then the experiment results are shown, followed by detailed evaluation and discussion.

5.1 Data Sets

Two data sets are used in the experiments, the first data set is the Special database II from the National Institute of Standards and Technology (NIST). This data set contains 12 different sets of tax forms of IRS 1040 Package X for the year 1988 totaling 5,590 IRS tax documents. This data set will be referred to as the NIST-IRS data set. These documents include Forms 1040, 2106, 2441, 4562, and 6251 together with Schedule A, B, C, D, E, F, and SE. These documents have 20 different form faces and are all binary images of consistent quality. These forms have a small pair-wise similarity as shown in Figure 1(a) and 1(b), in terms of both the image feature and the actual string contents they contain.

The other data set contains 4,576 event report forms from the US Food and Drug Administration (FDA). This document set consists of 12 different form faces, which are the actual event reports. These forms are also binary images scanned in different resolutions, from 199dpi to 300dpi. Documents following the same templates were produced at different times, with various fill-in data. Most of the documents are noisy with various skewing angles and annotation. Some of the templates are very similar in term of image layout and string contents, making these difficult to differentiate. Figure 1(c) and 1(d) illustrates a pair of similar templates.

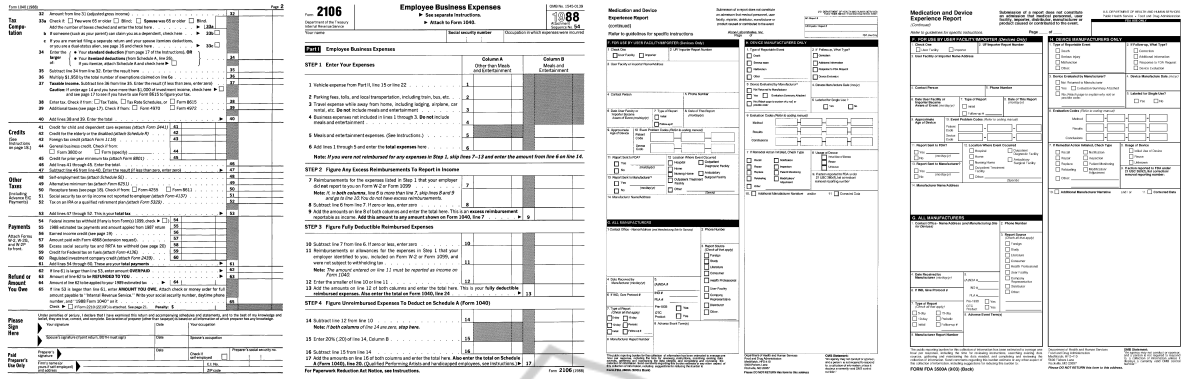
5.2 Comparison Methods

The detailed implementation and set up for the comparison methods, Cosine Similarity, SVD, and Zone-Seg are explained as follows.

5.2.1 Cosine Similarity

Given two documents templates T_i and T_j, all terms (i.e., strings) that occur on the whole collection are first collected to obtain the term-frequency vectors C_i and C_j, and their cosine similarity is computed with Equation 1.

As the term frequency cannot be negative, similarity between T_i,T_j is bounded in the interval $[0,1]$. When T_i and T_j have no common terms, $similarity\{T_i,T_j\} = 0$, and if T_i,T_j are exactly the same, then $similarity\{T_i,T_j\} = 1$. To use cosine-similarity to identify the template of a document, each document template is represented as a term-frequency vector $C_i(i = 1,2,\ldots,N_T)$. The dimension of these term frequency documents is the total number of distinct terms that appear on the whole set of templates. For each unlabeled test document E, the corresponding term frequency vector C_E is computed. The similarities between the test document and the templates $similarity\{E,T_i\}$ are computed. The template T_i with the largest cosine-similarity to the unlabeled test document is taken as the matching template.



(a) A NIST template (b) Another NIST template (c) An FDA template (d) Another FDA template

Figure 1: Sample NIST and FDA templates. Notice (a) and (b) are significantly different while (c) and (d) are quite similiar.

5.2.2 Singular Value Decomposition

The Singular Value Decomposition (SVD) approach is implemented as follows. All terms on templates $T_1, T_2, \ldots, T_{N_T}$ are used to generate the term document matrix. We have experimented with several common weighting functions. The two weighting functions which give the best accuracy were used in the experiment. The log weighting function is used as local weighting function. It is defined as: for term i in template T_j, the weighting $l_{i,T_j} = \log(c_i^j + 1)$, with c_i^j being the count of occurrences of term i in template T_j. The global weighting used is the Entropy weighting function, which is reported to work well with LSI. A term i's global weighting is computed as $g_i = 1 + \sum_j \frac{p_{i,T_j} \log p_{i,T_j}}{\log N_T}$, where $p_{i,T_j} = \frac{c_i^j}{c_i^{UT}}$, and c_i^{UT} is the total count of occurrences of term i in the whole collection UT. All ranks of singular values have been tested, and the half rank and full rank results are reported.

5.2.3 Image feature similarity / ZoneSeg

Each template T_i and the input document E are treated as an image, and grids are superimposed on the images with size $m \times n$. For each small patch of the image, if the black pixels on the patch exceed over a threshold $K_{threshold} = 5\%$, the patch is represented as 1, and 0 otherwise. Then, according to the same sequence, the image can be represented as a string consisting of 0s and 1s. The similarity between a template T_i and an input document E is defined with the Levenshtein distance between the two strings. The more visually similar the two images are, the smaller the Levenshtein distance will be. The Levenshtein distance between the test document E and each template T_i is computed, and the template that minimizes the distance is taken as the identified template. Different image patch sizes are also experimented with, and the size $m = 36$ pixels, $n = 24$ pixels, which renders the best results, is used in the experiment.

5.3 Experiment Design

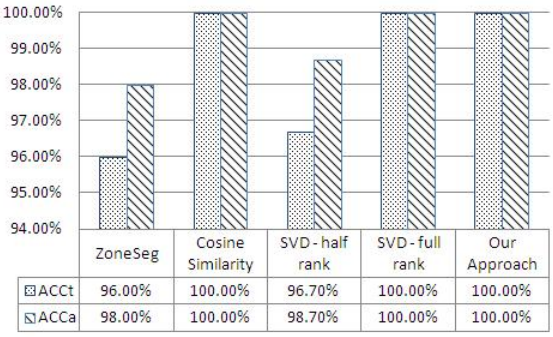
The experiment was conducted on a set of Intel Core Duo machines, using Python 2.7, and MatLab 2009b. For OCR, we used the open-source Tesseract 3.01 project.

5.4 Results and Analysis

In this section, we compare the performance of our approach with cosine similarity, SVD, and ZoneSeg. The performance is reported in terms of accuracy. We evaluate the accuracy of a document identification algorithm based on two metrics: total accuracy and average accuracy on each template. The total accuracy ACC_t of an algorithm is defined as the ratio of number of documents which are correctly identified and the total number of documents. The average accuracy on each template ACC_a is defined as the average accuracy on documents from each template. While the ACC_t accuracy demonstrate the overall performance of each method on the data sets respectively, the template-wise average accuracy ACC_a shows the algorithm robustness towards different templates. The corresponding results on NIST-IRS and FDA data set are shown in Fig. 2 and 3 respectively.

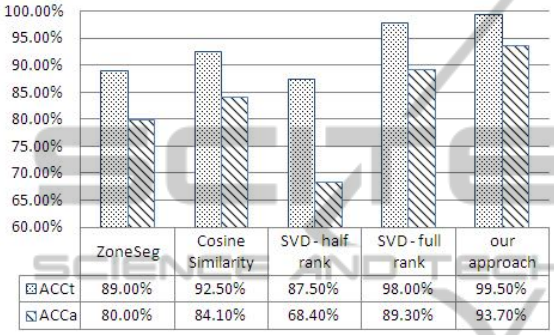
From Figures 2 and 3, we make the following observations.

1. All algorithms achieve good results on the NIST-IRS data set.
2. For the NIST-IRS data set, all algorithms achieve 100% ACC_t accuracy and ACC_a accuracy, as expected, except for ZoneSeg. However, the result



	ZoneSeg	Cosine Similarity	SVD - half rank	SVD - full rank	Our Approach
ACCt	96.00%	100.00%	96.70%	100.00%	100.00%
ACCa	98.00%	100.00%	98.70%	100.00%	100.00%

Figure 2: Performance on the NIST-IRS data set.



	ZoneSeg	Cosine Similarity	SVD - half rank	SVD - full rank	our approach
ACCt	89.00%	92.50%	87.50%	98.00%	99.50%
ACCa	80.00%	84.10%	68.40%	89.30%	93.70%

Figure 3: Performance on the FDA data set.

of ZoneSeg algorithm is still consistent with the reported results in (Esser et al., 2011).

3. For the FDA data set, our algorithm achieves the best ACC_t accuracy and ACC_a accuracy.
4. SVD methods performs better on full rank than on half rank on both data sets.
5. For the NIST-IRS data set, ACC_a result is better than ACC_t with all algorithms. For FDA data set, ACC_t result is better than ACC_a consistently for all algorithms.

The first observation is expected: the accuracy of document identification algorithms depend on the data sets. When the difference of different templates in the data set is significant, it is easier to identify the template. Hence, the accuracy will be high. Since the NIST-IRS data set's templates are significantly different both in image level and contents, high accuracy for most algorithms was expected.

The second observation indicates that the differences between different templates are significant. Both the algorithms based on OCR output and the algorithms based on image level features achieve good results, all being over (98%). However, the algorithms based on OCR output achieve better accuracy than algorithms based on image level features because the difference in string level features is more significant than the difference in image level features. The NIST-IRS data set also has less fill-in data in common, which makes the test images easier to be identified using image level features. However, the noise on the image, including the scanning skewing and translation, still affects the accuracy.

The third observation indicates that the proposed probability-based algorithm works very well even on the templates that are very similar. Because this approach considers both the reward on the evidence of an unlabeled test document being any of the templates, it also penalizes the dissimilarity with probability. Unlike other algorithms that mainly measure similarity between a query and a template, our model measures both the similarity and the dissimilarity between a query document and a template. The accuracy of cosine similarity and SVD methods is not as good as our algorithm because they do not have a penalty mechanism, and cannot consider the possibility of a match on a string occurring when the fill-in data coincides with the term used in another template. The accuracy of the ZoneSeg is lower on the FDA set, indicating that the performance of ZoneSeg is sensitive to the quality of the document and the similarity between templates.

The fourth observation indicates that full rank SVD works better than smaller ranks. As in most information retrieval task, usually only few ranks are used to approximate the data in feature space, which is accurate enough. However, a fewer rank does not approximate the full rank that well for this task.

The last observation indicates an important result. Since ACC_a is the average of the accuracy for each template, in the case that an algorithm performs consistently on all templates, the two metrics should give the same result. However, when the number of samples for each template differs, the accuracy of the same algorithm differs across different template detection result; so, the two metrics have different values. For the NIST-IRS data set, the accuracy ACC_a is consistently higher than the accuracy measurement ACC_t. This is because the accuracy of several templates, whose sample sizes are smaller, all algorithms still give a high accuracy, most of which is higher than the templates with larger sample size. However, for the FDA data set, there are a few templates with small sample sizes, and the accuracy for these templates are lower. The templates that are harder to identify have a smaller number of test documents. Note that our probabilistic approach has the smallest decrease in accuracy from ACC_t to ACC_a among all methods, which indicates that our method is more consistent across different templates.

We also studied the 23 failed cases out of the 4,576 tests in the FDA data set and found that the failed

cases generally fall into two situations: 1. The images are severely malformed (e.g., more than half of the document was not scanned), and only the identical portions of the images are kept and OCRed; 2. For a pair of templates, which share almost the same structure, having 10 or fewer different words, the test was unable to determine the template, given that the scores computed by Equation 5 for the pair were too close. Simply choosing the maximum score results in incorrect classification. To handle these cases, a threshold on the likelihood should be set based on application and similarity among the templates. Any test file whose likelihood is below this threshold should be verified manually.

With respect to speed, the experiments show that our method's speed is comparable to cosine similarity and SVD, and ZoneSeg has a varied speed depending on the size of the patch. In the experiments, the image patch size that gives the best accuracy is significantly slow, since a general test image will be converted into a string having a length greater than 5,000. This results in the Levenshtein distance computation to take a long time to execute.

6 CONCLUSIONS AND FUTURE WORK

A probabilistic method of identifying document templates for noisy scanned document has been studied. This method works well with low accuracy OCR results produced from noisy documents. Through experiment and analysis, the proposed method is shown to perform consistently across different template sets, and works well even when document templates are very similar. We recognize that certain documents will contain, in addition to text, non-text content for which our technique does not apply. It is the intent that the technique in this paper will be incorporated into larger application systems that handle both text recognition (our technique) and non-text recognition (image feature-based techniques).

Our future research will focus on incorporating other image features-based techniques with our methods and identifying fill-in data automatically based on the proposed method.

REFERENCES

Blei, D. M., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Cunningham, H., Maynard, D., Bontcheva, K., and Tabla, V. (2002). Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*.

Deerwester, S. Improving information retrieval with latent semantic indexing. In *Proceedings of the 51st ASIS Annual Meeting*, ASIS '88.

Esser, D., Schuster, D., Muthmann, K., Berger, M., and Schill, A. (2011). Automatic indexing of scanned documents - a layout-based approach. In *Document Recognition and Retrieval XVIII*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57.

Hu, J., Kashi, R., and Wilfong, G. (2000). Comparison and classification of documents based on layout similarity. *Inf. Retr.*, 2:227–243.

Jinhui Liu, A. K. J. (2000). Image-based form document retrieval. *Pattern Recognition*, 33:503–513.

Lu, Y. and Tan, C. L. (2004). Information retrieval in document image databases. *IEEE Transactions on Knowledge and Data Engineering*, 16:1398–1410.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press.

Salton, G. (1986). Another look at automatic text-retrieval systems. *Commun. ACM*, 29:648–656.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. In *Communications of the ACM*, volume 18.

Shin, C., Doermann, D., and Rosenfeld, A. (2001). Classification of document pages using structure-based features. *International Journal on Document Analysis and Recognition*, 3:232–247.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley.

T. S. Jayram, Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., and Zhu, H. (2006). Avatar information extraction system. In *IEEE Data Engineering Bulletin 29*.

Zheng, Y., Li, H., and Doermann, D. (2005). A parallel-line detection algorithm based on HMM decoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:777–792.