IJDAR (2012) 15:167–182
DOI 10.1007/s10032-011-0163-7

ORIGINAL PAPER

Genre identification for office document search and browsing

Francine Chen · Andreas Girgensohn · Matthew Cooper · Yijuan Lu · Gerry Filby

Received: 20 October 2010 / Revised: 9 February 2011 / Accepted: 22 March 2011 / Published online: 26 April 2011

Abstract When searching or browsing documents, the genre of a document is an important consideration that complements topical characterization. We examine design considerations for automatic tagging of office document pages with genre membership. These include selecting features that characterize genre-related information in office documents, examining the utility of text-based features and image-based features, and proposing a simple ensemble method to improve the performance of genre identification. Experiments were conducted on the open-set identification of four coarse office document genres: technical paper, photo, slide, and table. Our experiments show that when combined with image-based features, text-based features do not significantly influence performance. These results provide support for a topic-independent approach to identification of coarse office document genres. Experiments also show that our simple ensemble method significantly improves performance relative to using a support vector machine (SVM) classifier alone. We demonstrate the utility of our approach by integrating our automatic genre tags in a faceted search and browsing application for office document collections.

F. Chen (✉) · A. Girgensohn · M. Cooper · G. Filby
FX Palo Alto Laboratory, Inc., 3400 Hillview Ave,
Bldg. 4, Palo Alto, CA 94304, USA
e-mail: chen@fxpal.com

A. Girgensohn
e-mail: girgensohn@fxpal.com

M. Cooper
e-mail: cooper@fxpal.com

G. Filby
e-mail: filby@fxpal.com

Y. Lu
Texas State University, San Marcos, TX 78747, USA
e-mail: yl12@txstate.edu

Keywords Genre identification · Office documents · Image features · Text features · Classification

1 Introduction

Genre[1] is a popular way of categorizing movies, songs, and literature. Documents have also been categorized into genres such as fiction, opinion, brochure, slide presentation, technical document, blog, and home page. Document genres such as these can improve document search. While entering query terms is the traditional method of searching for documents, genre provides a complementary, non-topical means to characterize documents and web pages and is useful metadata for indexing, organizing, and searching for documents.

Documents occur in different modalities, including plain text, web, and imaged documents. In enterprise intranets and on the web, there are many *office documents*, which we define as digital documents created and shared by office workers. These documents are in formats such as PDF, PowerPoint, and Word. In this paper, we focus on automatic identification of four coarse genres of office documents: *technical paper, photo, slide,* and *table*. We define technical papers as research-type papers such as conference papers, journal papers, and reports. Photos are pictures taken with a camera and may be embedded in documents, but do not include drawings or screenshots. Slides are defined as any page or document created for presentation in front of audience. Tables include tables from spreadsheets as well as tables in papers and other documents.

Although document genre is related to file format, they are not identical. A document creation tool, such as Word

[1] Merriam-Webster: a category of artistic, musical, or literary composition characterized by a particular style, form, or content.

Fig. 1 Sample document pages from the genres of (**a**) paper (**b**) photo (**c**) table (**d**) slide

or PowerPoint, can be used to create documents in several genres. For example, in addition to its primary use in creating slides, PowerPoint is also used to create figures and drawings for papers and also to create handouts of screen shots and photos. And some extensions, such as PDF, are associated with many genres, since files created in different formats are often converted to PDF for its portable representation. Thus, if a user were looking for slides from talks and looked only at PowerPoint files, the user may be presented with extra files that are not slides; furthermore, the user will not see slides that are in other formats, such as PDF.

Layout has been useful in genre identification of scanned documents (e.g., [1,10]); it is encoded in the markup of web documents and has been found to convey strong cues about web genre (e.g., [27]). In office documents, one cannot assume that all documents contain encoded layout information. For example, some PDF documents are created from scanned documents. Furthermore, when the information is present, it may be inconsistent, e.g., not all PDF documents correctly encode column information. We thus identify genre from imaged document pages that have been OCR'd, so that both layout and text information are available for all pages; this also obviates the need to determine whether layout information, when given, is correct.

Examples of the four office genres of (technical) paper, photo, table, and slide are shown in Fig. 1. These examples illustrate some of the diversity within genres and also the ambiguity of the genres. Note in Fig. 1b, the photo category is composed of not only photos taken from a camera but also document pages that are mostly photo (the top-right and the middle-right examples). Note also that the bottom-left slide in Fig. 1d is composed mostly of photos. Thus, it can be appropriate for some pages to be tagged with more than one genre. We tag a page with a genre when at least half the page represents the target genre. With this definition, the middle-right page in Fig. 1c is tagged as a table and as a paper, and the lower-right page is tagged as a table. Thus, tables that also contain text as footnotes and other information are tagged as tables.

While web search strongly relies on link information for ranking, office documents contain very little link information, making genre identification more valuable when searching office documents. The ability to automatically identify office document genre can allow search results to be grouped by genre or topical search queries to be augmented by genre. With our four proposed coarse genres, one could specify that technical papers on geysers are of interest, or if one was looking for a brochure, specify that none of the genres are of interest. These genres complement a number of finer genre classifications that are characterized primarily by differences in textual content. Examples of such genres include legal, scientific, non-fiction and fiction (e.g., [16]) and Editorials, Letters to the Editor, Reportage, and Spot News (e.g., [31]). Our office genres also complement web genres, such as home pages, blogs, and wikis.

For genre identification, we assume that the various page types that occur in a particular office document genre should be learned by the classifier. Thus, rather than classifying into page types such as title, table of contents, references, etc., we classify each page by genre. This eliminates the need to create classifiers for each possible page type and the need to map between page types and genre. For text-based genre classification, Kessler et al. [16] distinguish between "surface" cues, which are simple to extract, and "structural" cues, which are related to linguistic structure. Here, we define a set of document-inspired, surface-type image features. It has been suggested that there are advantages to performing genre identification without text features, including language independence [27] and corpus independence [2]; image features may have these advantages if they perform well without text features. Since some documents belong to more than one genre, our genre identifier is open set, so that a document is tagged with zero or more genres. We then extend our *page* genre identification to predict *document* genre membership. The automatic genre identification results are used in a system developed for faceted search and browsing of enterprise collections that offers genre as one of the possible search facets.

The contributions of this paper center on an investigation of methods for office document genre identification. These include: (1) developing a set of document-inspired "surface" image features and experimentally comparing their utility

to text-based features, (2) adding a simple ensemble method that efficiently makes use of the data to improve genre identification performance, (3) developing a document genre identification approach where the different page types associated with a genre are learned by the classifier, and (4) deploying the results in a faceted search and browsing system.

The rest of the paper is organized as follows. In Sect. 2, we review earlier text-based and image-based genre identification work and compare them to our approach. We also discuss current techniques for genre identification of web documents and why those techniques do not directly apply to office documents. In Sect. 3, we describe the data set of representative office documents that we collected and labeled for this work. In Sect. 4, we describe our proposed text-based and image-based feature sets, and the classification methods used, including a simple ensemble method. In Sect. 5, we present experimental results that show the utility of our image-based method over a text-based approach. In Sect. 6, we discuss the combination of page genre scores to predict *document* genre, and then we illustrate the use of our genre tags in a document searching and browsing system. Conclusions and future directions are presented in Sect. 7.

2 Related work

Many sets of genre categories have been proposed for text genre identification and web genre identification. For web page genre, Roussinov et al. [24] conducted a study of useful genres and proposed five major web document genre groups, while Santini et al. [25] compiled and published more than 80 genres.[2] Henderson [12] conducted a study analyzing the computer folder structure of six knowledge workers and found genre to be the most common organizing factor. She also states that "people may deal with a vastly differing set of genres, depending on their job". Thus, the set of useful genres depends, in part, on the planned application.

In our work, we focus on automatic genre identification of the types of documents used in a research lab, which are a subset of those found in an enterprise. Our genre set can be further refined to application-specific genres of interest using text-based methods.

Different modalities of features have been defined for use in genre identification. In works on genre identification based on textual features, e.g., [7,16,19,31], the features are primarily computed from *surface* cues, such as document terms and punctuation, or *structural* cues, such as part-of-speech tagging. Kessler et al. [16] found that performance using surface cues was similar to that using structural cues and recommended surface cues because they are much easier to compute. Stamatatos et al. [31] noted in an experiment on genre identification that using less than 50 of the most frequent terms in English, which are primarily stop words, plus punctuation, as "style" features was effective (greater than 97% accuracy) for classifying a subset of the Wall Street Journal text corpus into four genres. Lee and Myaeng [19] examined frequency-based methods for identifying terms that occur across genres and also across subject, i.e., topic, classes within a genre. The use of subject classes improved genre classification performance when used with a similarity-based approach, but when used with a Naïve Bayes classifier, the best performance was when subject classes were not used. Since we take a classification-based approach, we do not use subject classes in our text-based genre classifier. As in [19,31], we use surface cues, but in addition, we use discriminative selection of text features.

Web genre detection commonly employs features that describe markup attributes of web pages, in addition to analyzing the pages' textual content. Link plus formatting or layout tags have been used, as well as web page URLs (e.g., [2,5,23]). Meyer zu Eissen and Stein [5] created a web genre identification system and found HTML markup to be more useful than text features. Levering, et al. [20] found that adding HTML features to text features improved performance. They also noted that adding visual features derived from the HTML tags sometimes helped and sometimes hurt performance, depending on genre. Recently, Scholl [27] presented a system for web genre detection that used *only* features derived from HTML markup and obtained relatively good accuracy. Thus one may ask whether features capturing layout in office documents are similarly more effective than text-based features for genre identification of office documents.

One approach that addresses the absence of encoded layout information in office documents is to perform image-based layout analysis (e.g., [1,10]). Each page is segmented into zones that are labeled with tags such as text body, picture, and table. The tagged zones are then used to classify pages according to genre. As can be observed in a survey paper by Chen and Blostein [4] on classifying "mostly-text" document page images into a variety of types, the use of layout is predominant in document page image classification approaches. However, as noted in [4], layout analysis can be complex and error-prone.

A few prior works have explored genre identification based on image analysis without layout information. Das Gupta and Sarkar [8] describe a genre classifier based on identified salient feature points for discriminating between two genres, journal articles and memos, and tested on 80 page images. Shin et al. [28] used window-based features to identify page type, i.e., title page, cover page, reference page, table of contents, and form. They define over 20 "structure-based" features that include number of text column gaps, configuration of lines on a page, and classification of each

[2] http://www.webgenrewiki.org/index.php5/Genre_Classes_List

window into the region types of text, image, or graphics. Analogous to the surface and structural types of cues for text features, these features may be thought of as structural cues, while simple visual features that can be extracted directly without classification are analogous to surface cues. We also use window-based features, and some of our features attempt to capture similar document characteristics. However, our features are primarily surface features and our goal is page and document genre identification, rather than page-type identification.

Most previous genre identification systems make use of either image-based or text-based features. Kim and Ross [18] compared image-based and text-based features for genre identification. For image-based features, they used simple, surface-based features in tiles from the *first* page of a document to identify document genre. Kim and Ross also investigated the use of two types of text-based features. One is based on "prolific" terms, and the other is based on significant topical terms. We initially planned to use their image-based system and their prolific terms-based system as baseline systems. We did not consider the topical terms feature because we wanted features that are topic-independent, and as mentioned earlier, other researchers achieved good results without topical terms. However, the prolific terms feature was problematic for our genre set, since two of our genres did not have any prolific terms as defined in [17]. We then decided to use Stamatatos et al. [31] as a baseline text system because their system demonstrated good accuracy. Kim and Ross's image-based system was used as an image baseline.

As with the Kim and Ross image-based system, we forgo layout analysis and use tiles. However, in contrast, we develop a set of document-inspired features that attempt to capture layout characteristics and texture of the tiles without requiring region classification. Our work further differs from Kim and Ross in two ways. First, we examine the utility of combining feature types. Second, we develop an open-class system for identifying a set of genres.

Most prior image-based genre identification approaches focus on page genre identification. Those that perform document genre identification commonly use only the first page of a document (e.g., [1,18]), which is the most distinctive page for some genres. In contrast, we identify the genre for each page separately and then combine the page genre tags to identify *document* genre membership. With our approach, search and browsing can be performed based on document genre or page genre.

In summary, there has been much work on text, image, and web page genre identification. Also, HTML-based tags have shown to be very useful in differentiating web genres. There has been much less work on differentiating the genres of office documents. In this paper, we show how the genre of office documents, which commonly do not have link or tagged layout information, can be identified using surface image-based features. We also show that our text-based features, while exhibiting better performance than a baseline based on [31], are not as useful for identifying office document genres. We improve performance using a simple ensemble method and combine per-page genre labels to classify document genre.

3 Office document data set

A benchmark data set of office documents that has been labeled with our target genres, *slides, technical papers, tables* and *photos,* does not exist. To create a corpus, we collected a set of documents and tagged them with our four target genre types. The documents were collected primarily from the web, with some documents also added from our corporate intranet. We queried Google for 100 results per query, scraped the returned URLs for office document extensions, such as .pdf, .xls, and .doc, and downloaded those files. The queries included a number of document genres (e.g., table, brochure) as well as topical terms covering a variety of subjects (e.g., airline, baby, car, disease, dog, plants, semiconductor, shipping, software, stocks, vacation). We also added photos obtained from Flickr and about 100 documents from our corporate intranet. Note that in addition to the Flickr photos, photos also occurred in some of the other documents, such as presentations and magazine articles.

In order to uniformly process all documents, each page of a document was represented as a JPEG image. We checked for duplicates, manually looking for those with identical or almost identical content, and removed them. The genre-based queries identified documents both within and outside the desired genre. For example, 'table' contained document data tables, as well as periodic tables, and pages containing other senses of the word, such as "table tennis", "table eggs", "table linens", and the verb in "table a motion".

Eleven people participated in labeling the corpus pages with our four target genres. The labelers used an interface we developed for quickly selecting pages belonging to a specified genre. All pages of the documents in a folder were displayed in a scrollable window and the pages could be selected using common interface selection techniques.

Each labeler was instructed to find and mark pages for which one given genre (or one genre at a time for the person who labeled two genres) was apparent in at least half of the page. With this procedure, there may be pages with multiple genre labels and pages that are not labeled as any genre.

Labeling pages for a selected genre was sometimes ambiguous. For example, a table of contents usually has two columns, but we decided not to include these as 'tables' because they are more closely associated with genres such as reports and books, rather than generic tables. Another example is

Table 1 Statistics of the office document data set

Genre	# pages	# documents
Paper	948	87
Photo	541	447
Slide	1,765	120
Table	681	144
Total tags	3,935	798
Corpus	5,098	1,178

Table 2 Labeler agreement

	Paper	Photo	Slide	Table
Fleiss' kappa	0.88	0.81	0.97	0.92

that the "boundary" defining technical papers is ambiguous, since some magazines and newsletters are on technical topics. To resolve this, we decided to define technical papers as research-type papers such as conference papers, journal papers, and reports.

The corpus contained 1,267 pages with no tags. As indicated by the ambiguous examples above, some of the pages without tags were relatively similar in layout to our four target genres. There were also 3,727 pages with one tag and 104 pages with two tags. The center column of Table 1 shows the number of pages in the corpus tagged with each of the four genres. There were a total of 3,935 tags assigned to 5,098 pages.

Table 2 shows the agreement among our labelers for each genre, as measured by Fleiss' kappa [6]. Fleiss' kappa measures the extent of labeler agreement above chance when there are more than two labelers. It ranges in value from negative (poor agreement) to 1.0 (complete agreement). The agreement among our labelers was relatively good. However, there was some disagreement, indicating ambiguity in the page genres.

For our experiments, the labeled data were partitioned by document into three sets with equal numbers of documents (ignoring round-off). The partitioned data were used for performing 3-way cross-validation.

4 Genre identification

In our approach to genre identification, the different page types associated with a genre are learned by the classifier model. That is, we do not create separate models for each page type, such as title page, table of contents, and reference pages, which are all page types associated with the technical papers genre. This is reflected in the instructions to the labelers to select pages of a given genre, rather than page type. With these genre labels, we create one one-versus-rest model per genre.

In the rest of this section, we describe the features and classifiers we use for page genre identification. We also propose a simple ensemble method for improving genre identification performance.

4.1 Features for genre identification

We developed a set of image-based features to capture document layout characteristics. We also developed a set of text-based features that combine ideas proposed in earlier works and incorporate feature selection.

4.1.1 Image-based page genre features

As an alternative to performing layout analysis of page images, we tile each page image and extract a set of features that capture local document characteristics, such as lines of text, *within a tile*. While this requires relatively large tiles, the tiles need to be sufficiently small to distinguish the different region types (e.g., heading, figure, body text). Empirically, we have found that dividing each page image into a grid of 5 tiles horizontally by 5 tiles vertically, for a total of 25 tiles, meets our requirements. A full "page" tile is also used to capture features that may span multiple tiles, such as table rule lengths. Figure 2 illustrates the tiling of a page.

Our feature set was designed to differentiate the types of regions identified in layout analysis, such as text, images, rules, and columns. The tile features are:

Image density. This feature is similar to the image feature used by Kim and Ross [18], except our tile size is much larger. This feature attempts to capture areas where print is sparse and should help differentiate titles, regular text, and photos. To compute it, a page image is converted to a binary image and the ratio of the number of black pixels to the total number of pixels in each tile is calculated.

Horizontal projections. This feature attempts to capture the number of text lines and the distribution of text line heights. This helps differentiate genres such as slides, which are of a larger font, and photos, which have few, if any text lines. The feature can be computed based purely on image processing techniques (e.g., [33]) to project the pixels horizontally once the text foreground is identified. Alternatively, word bounding box locations, e.g., provided by an OCR system, can be used to compute the projections of the bounding boxes. We used the latter approach because each page was OCR'd for the text features. From a projection, the peak widths, roughly corresponding to text rows, are quantized into a five-bin histogram. This characterizes the number of rows and the distribution of font sizes.

Vertical projections. This feature attempts to capture the number of text columns and the distribution of their widths.

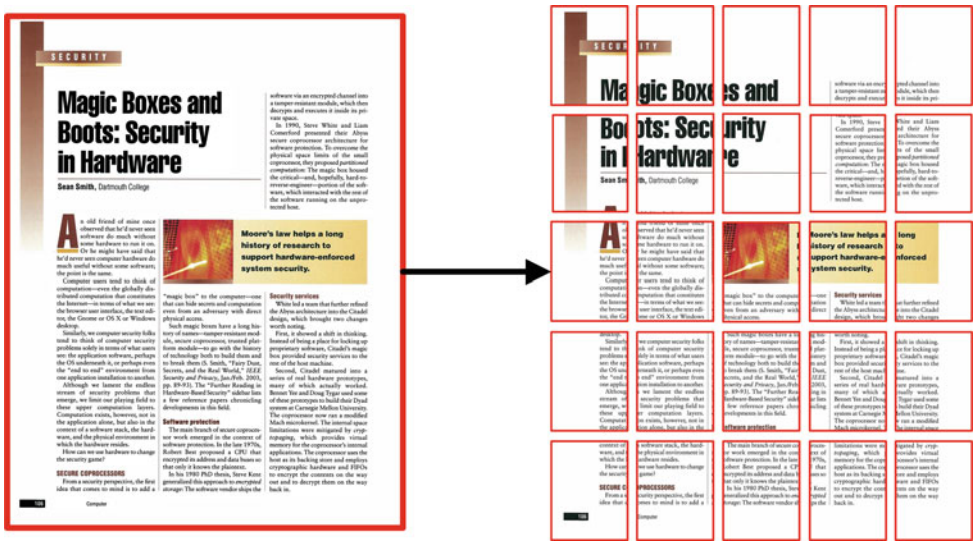

Fig. 2 Tiling a page

This feature can help to differentiate papers and tables, which may have columns of varying widths. As with the horizontal projections, we project word bounding box locations and compute a five-bin histogram of column widths.

Color correlogram. Color correlograms represent the spatial correlation of colors in an image and have been used for image retrieval [14]. We select a subset of the correlogram values to capture texture and color variation. To compute this feature, the images are proportionally scaled to a maximum of 1,550 pixels in the horizontal and vertical directions, and then a color autocorrelogram computed. The colors were quantized to 96 colors and distances of 0, 1, and 3 pixels were used, resulting in 288 dimensions per tile. To reduce the number of correlogram coefficients, feature selection was performed using the Maximally Relevant Minimally Redundant (mRMR) feature selection method [22] to identify a subset of 50 features. Because the feature values depend in part on tile location (e.g., titles usually appear near the top of a page), the feature selection method performs better when implicit spatial information is preserved. This was done by concatenating the correlogram coefficients for each tile in a page into a vector and then performing feature selection over all concatenated correlogram vectors in the training set (Fig. 3b). Our approach contrasts with the "bag of visual words" approach to image classification [29], where the locations of the image tiles are not preserved. To enable modeling of all region types in any tile, the final feature set is composed of the union across tiles of the selected feature positions (Fig. 3c). Then for each page, the selected features for each tile are concatenated in a fixed order (Fig. 3d, e). With the tile locations implicitly encoded in the feature vector, a genre classifier can learn the feature values observed in each location.

The concatenated vector is combined with page-based features to create a feature vector representing a page. Two types of page-based features are computed:

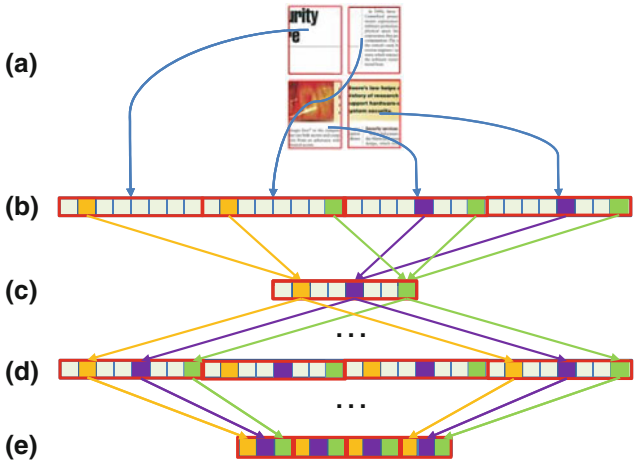

Fig. 3 Feature selection over tiles (each tile is outlined in *red*, each square in **b**–**e** is a feature): **a** Image tiles **b** Correlograms of tiles are concatenated into a feature vector. mRMR-selected features are *colored*. **c** Union of selected feature positions, the *colored* features form the feature set for all tiles. **d** Features to be extracted from concatenated correlograms are *colored*. **e** Final feature set

Lines. Tables often have many rules which extend horizontally and/or vertically. To identify rules on a page, run lengths of black pixels in a binary image are computed horizontally, allowing for short pixel jogs horizontally or vertically [33]. The number of lines in each tile is noted and the line lengths are uniformly quantized into a four-bin histogram. This differentiates lines that span a page width, a column, and shorter lines. Vertical line histograms are computed similarly.

Image size. The width and height of the image. Technical papers and slides commonly occur in standard page sizes. Photos and tables, on the other hand, may be cropped and also may occur in document pages and slides.

Although we use feature selection on the entire set of text terms as described in the following section, we do not use

it on the entire set of image features. For our image-based features, the number of dimensions is relatively small, e.g., five-bin histograms per feature type, and each of the features is informative relative to the others. For example, the number of rules is independent of the number of lines of text. The exception is the color correlogram, where feature selection was employed as described above.

4.1.2 Text-based page genre features

Prior to extracting text-based features, the page images are OCR'd with Microsoft Document Imaging[3] and then preprocessed. The recognized text, which contains errors and extraneous characters, was tokenized into terms and punctuation. Some token types were mapped to a unique term representing that type; these token types were: integers, floating point numbers, underscores, lines, and dates of the form 'dd/dd/dddd' where 'd' is a digit. To reduce the number of terms due to OCR errors and to increase robustness, only terms that occurred in more than ten documents were kept. Stop words and punctuation were kept, since these tokens can be indicative of genre [16,23,31].

Previous text-based genre identification systems by [18] and [31] select frequent terms as features. "Prolific" terms that occur in many of the documents in each genre were used as text features in [18] to capture writing style. A fixed set of frequent terms, which corresponds to a subset of stop words and punctuation, was used by [31]. The features in these systems are primarily stop words and were motivated in part by the desire to select non-topical terms. However, these earlier methods do not consider whether the terms are good at discriminating among classes. For example, a frequent term may occur at approximately the same frequency in each class and therefore would not provide useful information for classification.

Non-discriminative selection methods, such as those used by [18] and [31], are suboptimal compared to Mutual Information (MI) and χ^2 based methods [21], which are popular feature selection methods. Since these feature selection methods are greedy, redundant features may be selected. For our text-based classifier, we used a discriminative feature selection method that is an extension of the MI-based selection method. The Maximally Relevant Minimally Redundant (mRMR) [22] method selects features that discriminate the different classes but minimize redundancy among the retained features.

We performed preliminary experiments using mRMR to select 50 terms per genre, but the results were poor. We hypothesize that this may be due, in part, to the sparseness of text in some documents. To address the sparseness problem, we next tried selecting 100 and 200 terms, which seemed to be the limit of the number of terms mRMR could select from our office document data set in a reasonable amount of time. The results with 200 terms were much better, and so we select 200 terms per genre.

[3] http://office.microsoft.com/en-us/help/HP030763951033.aspx

Table 3 Top terms selected by mRMR for one data partition

Paper	Photos	Slides	Tables
the	photos	the	ltd
figure	skiing	these	comparative
however	to	has	west
both	peru	this	inform
these	racetrack	first	ge
this	gourmet	are	time
shown	1388-10	was	1-2
such	viewer	one	pesticide
between	napa	have	japanese
results	systems	that	stanford

For performing cross-validation experiments, the data were partitioned as described in Sect. 3 and term selection was performed separately for each partition. The selected terms for a partition form the text features for all partitions when that partition is used as the training set.

For each of our four target genres, the top ten terms selected by mRMR for one data partition are shown in Table 3. Note that the 'Paper' genre includes the content words "figure", "shown", and "results", in addition to stop words. In contrast, the top mRMR features for the Slides genre are primarily all stop words, and the top mRMR features for the Tables genre are all content words. Note also that terms are present for the Photo genre. Although a few photos contain text that was OCR'd, the terms are also from pages that are mostly photo, as described in Sect. 3 and illustrated by two examples in Fig. 1b. We also used mRMR to select the maximally relevant (*not* minimally redundant) terms for the four genres and observed that the terms were primarily stop words.

For each data partition, we combined the set of 200 terms selected using maximum relevance and 200 terms selected using mRMR for each of the target genres as the basis for the text feature vectors. Our text features contrast with Stamatatos, et al. [31], who found that using less than 50 of the most frequent terms provided good genre identification for their corpus. Including the maximally relevant terms provides some redundancy, which may be helpful for genres where text is sparse. It also adds primarily stopwords and punctuation, capturing some of the traits of the style features in [18,31]. The text feature vector for each document is then composed of the term counts for each of the selected terms.

4.2 Classification

For most search systems, full coverage of all possible genres is not realistic. Instead, a genre identification system should be able to spot a subset of genres from a large possible set. For this, we create for each genre of interest a one-versus-rest model, where a classifier attempts to differentiate between the genre of interest and the "rest" of the genres. As noted earlier, each classifier learns the variations in page types associated with a genre.

Based on the competitive performance of SVMs (Support Vector Machine [3]) for many classification tasks (e.g., [5,15]), we used an SVM classifier. As commonly recommended for better SVM performance, the feature vector components are normalized. We use a normalization that scales and shifts the component values to range between 0 and 1, as recommended in [13]. Each vector element, x_i is normalized using:

$$x_{i,\text{norm}} = \frac{x_i - \min_j(x_j)}{\max_j(x_j) - \min_j(x_j)} \tag{1}$$

where $\{x_j\}$ are the observed values for one vector element in the training data; the same scaling factors are applied to both the training and test data.

An SVM classifier is a supervised, discriminative classifier that computes a hyperplane that best separates two classes: $\mathbf{w} \cdot \mathbf{x} - b = 0$ where $\mathbf{x}$ is a set of points from a training set. w and b are chosen to maximize the margin between the two classes. By applying the kernel trick [3] to the dot product, a linear SVM classifier can be transformed into a non-linear classifier. This may allow for better handling of the non-linear feature space with multiple subtypes within a genre. For our experiments, we used SVMlight [26], which allows specification of different kernels and cost-factors during training. In particular, we performed a parameter sweep with linear, polynomial, and radial basis kernels, with the order of the polynomial kernels ranging from 2 to 4. We also swept the cost-factor from 1 to 15, biased toward detection of a genre. Since an optimal model is found for each genre and data partition, the best kernel and parameter values varied for each trained model.

For our experiments, we performed 3-fold cross-validation on the data. For each train/development/test combination, one data partition was used for training the SVM, and a second data partition was used to tune the genre models, selecting parameters to maximize the balanced F-score, F_1:

$$F_1 = \frac{2PR}{P+R}, \tag{2}$$

the harmonic mean of precision, P, and recall, R. These measures are defined [21] as:

$$P = \frac{tp}{tp + fp} \tag{3}$$

$$R = \frac{tp}{tp + fn} \tag{4}$$

where tp is the number of true positives (both the system and the ground truth tagged a page with a given genre); fp is the number of false positives (the system but not the ground truth tagged a page with a given genre), and fn is the number of false negatives (the ground truth but not system tagged a page with a given genre). After tuning the genre models, the "optimal" model for each genre was used to classify the page images in a third data partition.

Using the features and classifiers we have described, we created systems to compare experimentally: image-based features with the Weka [32] Naïve Bayes classifier (**imgfeats.NB**), image-based features with an SVM classifier (**imgfeats.SVM**), and text-based features with an SVM classifier (**txtfeats.SVM**).

4.3 A simple ensemble classifier

To improve performance, we explored a simple ensemble method that makes better use of the data partitions for training and tuning our SVM-based system. We initially considered using a traditional classifier combination method that uses the results from different types of classifiers on the same data. But in preliminary studies comparing different classifiers, SVM performance was much better than the other classifiers that we tried—Naïve Bayes, k-Nearest-Neighbors, and Random Forests [32]. Consequently, we decided instead to use an SVM classifier trained on different partitions of the data and combine the results. In contrast to bagging, which avoids overfitting by selecting exemplars with replacement to form multiple training sets, we divide our training data into two mutually exclusive partitions to train separate classifiers. We chose this approach to minimize the number of models that need to be trained.

In particular, the data set was divided into three partitions, A, B, and C, for cross-validation. The results from training on A, tuning on B, and testing on C to produce binary genre classifications g_1 were combined with the results from training on B, tuning on A, and testing on C, to produce binary genre classifications g_2. Thus, each test partition is labeled using two SVMs. Assume that the classification for genre k of page i by classifier c is $g_c^k[i]$. To create a higher precision classifier, we used this combination rule for the ith page:

$$(g_1^k[i] \vee g_2^k[i]) \wedge (\overline{g_1^m[i] \wedge g_2^m[i]}) \quad \forall m \neq k.$$

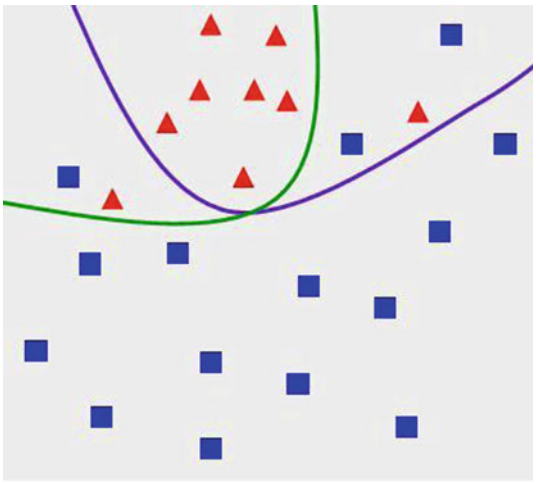

Fig. 4 Illustration of two SVM classifier hyperplanes trained on different data partitions

It requires that at most one of the classifier pairs in a competing genre would label the page as the competing genre.

An illustration of the intuition behind this method is shown in Fig. 4. Two classifiers trained on different data partitions identify different hyperplanes and have a portion of the space in common. We hypothesize that the common space is primarily pages of a given genre, and that the space covered by only one of the classifiers are pages that may be a combination of genres or a page that is otherwise ambiguous or unusual. When used with image-based features, we denote this system **imgfeats.Ens**.

4.4 Combination of text and image features for page genre identification

We considered whether using a combination of text and image features can improve performance for our genre classification task. In video analysis, Snoek et al. [30] compared "late fusion", which combines output from separate text-based and image-based classifiers, with "early fusion", which combines text and image features before classification. They observed mixed results across the set of classification concepts and suggested that the fusion strategy should depend on the concept to be classified. Since we were concerned about the sparseness of text in many of the photos and some of the slides, we chose to use "early fusion" and combine the text and image information at the feature level. The features were used in a simple ensemble classifier. We refer to this system as **txt+imgfeats.Ens**.

5 Experiments and discussion

We conducted experiments to investigate the effectiveness of the features and classification methods we have proposed for office document page genre identification. We first compared the utility of our text and image features against baseline features and against each other. We then tested the effectiveness of our partition-based ensemble classification method and of the combination of text and image features.

The pages in our office document data set may be labeled with zero, one, or multiple genres. For evaluation, we measure performance by precision, recall, and $F1$-score (see Eqs. 2–4), computing the macro-average across genres. The macro-average gives equal weight to each genre, while the micro-average gives equal weight to each page [21]. We did not use micro-average measures because we did not collect documents for our corpus in a way to insure that the distribution of genres is representative of the distribution of office documents on the web. In our experiments, the significance of observed differences in performance was tested using the paired Wilcoxon signed rank test.

5.1 Baseline image feature comparison

Kim and Ross' image-based system [18] was used as a baseline because their system is based on surface features in tiles and was developed for classifying document genres. We implemented their image-based feature extraction: each page was divided into a 62×62 grid, and each tile in the grid with at least one pixel of value less than 245 was assigned the value of '0'. The other tiles were assigned a value of '1'. For classification, Kim and Ross compared Naïve Bayes, Naïve Bayes with kernel density estimation, Random Forest, and SVM classifiers using the Weka software package [9]. They observed that the SVM classifier performed much worse on their image feature data than the other classifiers. Hence, for the baseline image feature experiments, the Kim and Ross image features were tested with the Weka Naïve Bayes classifier using both the plain and the kernel density estimation versions, and the Weka Random Forest classifier. These systems are referred to as KR.NB, KR.NBK, and KR.RF, respectively, and form our image feature baseline models.

The page genre identification performance for these baseline models is shown in the top three lines of Table 4. Similar to the image feature results in [18], where two different data sets were used, both Naïve Bayes models performed better than Random Forests overall. However, the observed performance was not significantly different whether or not kernel density estimates were used with the Naïve Bayes classifier. These results indicate that our office document data set is a challenging collection for page genre classification.

The last row of Table 4, labeled imgfeats.NB, shows the performance of our document-inspired image features (described in Sect. 4.1.1) with the Weka Naïve Bayes classifier. The overall performance as measured by $F1$-score of our features was 0.7240 versus the best Kim and Ross $F1$-score of 0.5326. In all cases, our document-inspired image features

Table 4 Performance of our image-based page genre features against Kim and Ross baseline image features with different classifiers

Classifier	Precision	Recall	$F1$-score
KR.NB	0.4161	0.8085^{r}	0.5272
KR.NBK	0.4169	0.8162^{r}	0.5326
KR.RF	0.6033^{nk}	0.4433	0.4885
imgfeats.NB	$\mathbf{0.6431}^{nk}$	$\mathbf{0.9406}^{nkr}$	$\mathbf{0.7240}^{nkr}$

Superscripts 'n', 'k', 'r', and 'm' indicate statistically significant improvement at the 0.05 level over KR.NB, KR.NBK, KR.RF, and imgfeats.NB models, respectively. The best performance in each column is in bold

perform significantly better than the Kim and Ross features [18]. In the rest of the experiments, our document-inspired image features are used as the image features.

5.2 Baseline text feature comparison

We use the text-based genre features proposed by Stamatatos et al. [31] for our text baseline. These features are corpus independent and based on frequent terms. More specifically, we used the 30 most frequent stopwords and eight frequent punctuation marks specified in [31] as features. These features were used with the same SVM classifier as in our text-based method with feature selection. We refer to this baseline system as **freqfeats.SVM**.

Table 5 compares the page genre identification performance of the frequent terms-based baseline (freqfeats.SVM) and our text-based method with feature selection (txtfeats.SVM, as described in Sects. 4.1.2–4.2). Note that the overall performance measure, $F1$-score, for the frequent terms was 0.4940 and for the feature selection method was 0.7470. While there was no significant difference in recall, our method employing feature selection performed significantly better than the frequent terms-based method in precision and $F1$-score. These results indicate that using only frequent terms does not provide a rich enough set of features for identifying our target genres.

Although we do not show significance for results across Tables 4 and 5, we observed that our text-based features, txtfeats.SVM, exhibited significantly better $F1$-score than any of the Kim and Ross image feature classifiers, but no significant difference in $F1$-score with our image features, imgfeats.NB.

Table 5 Performance comparison of text features using SVM-based genre classifiers

Classifier	Precision	Recall	$F1$-score
freqfeats.SVM	0.4146	**0.7743**	0.4940
txtfeats.SVM	$\mathbf{0.7458}^{f}$	0.7718	$\mathbf{0.7470}^{f}$

Superscripts 'f' and 't' indicate statistically significant improvement at the 0.05 level over freqfeats.SVM and txtfeats.SVM models, respectively. The best performance in each column is in bold

5.3 Image features and ensemble classification performance

We examined the performance of our image features using the classification models described in Sect. 4. Table 6 shows the performance of our image features with an SVM classifier (imgfeats.SVM), image features with our ensemble classifier (imgfeats.Ens), and the use of both image and text features with our ensemble classifier (txt+imgfeats.Ens). Comparing the image-based systems against our text-based method (txtfeats.SVM), Table 6 shows that for the three evaluation measures of precision, recall, and $F1$-score, all methods with image-based features had statistically significant better performance than our text-based method. From this, we infer that image-based features can be more useful than text-based features for detecting some office document genres.

While it is not indicated in the tables, we also observed that the use of our image features with an SVM classifier, imgfeats.SVM, had significantly better performance, as measured by $F1$-score, over use of our image features with a Naïve Bayes classifier, imgfeats.NB (from Table 4). Thus, the ensemble method was applied to the SVM classifier results. Examining the addition of our simple ensemble classifier, imgfeats.Ens, we note statistically significant better $F1$-score performance over imgfeats.SVM.

Although our text-based classifier did not perform as well as our image-based classifiers, we examined whether the combination of our text-based and image-based features enhances performance. The last two lines of Table 6 compare the combined features (txt+imgfeats.Ens) to the image features with the ensemble classifier method (imgfeats.Ens). Although precision improved and recall decreased when text features are added (txt+imgfeats.Ens), overall, there was no significant difference, as measured by $F1$-score.

We also examine $F1$-score performance by each of our methods for each target genre. Figure 5 shows that our image features (imgfeats.SVM) perform better than using text only features for all test genres. Both ensemble classifiers (imgfeats.Ens and txt+imgfeats.Ens) outper-

Table 6 Performance comparison after adding an ensemble classifier and after adding text features

Classifier	Precision	Recall	$F1$-score
imgfeats.SVM	0.8619^{t}	0.8708^{t}	0.8639^{t}
imgfeats.Ens	0.9464^{ti}	$\mathbf{0.8790}^{tb}$	0.9107^{ti}
txt+imgfeats.Ens	$\mathbf{0.9724}^{tie}$	0.8626^{t}	$\mathbf{0.9135}^{ti}$

Superscripts 't', 'i', 'e', and 'b' indicate statistically significant improvement at the 0.05 level over txtfeats.SVM, imgfeats.SVM, imgfeats.Ens, and txt+imgfeats.Ens models, respectively. The best performance in each column is in bold

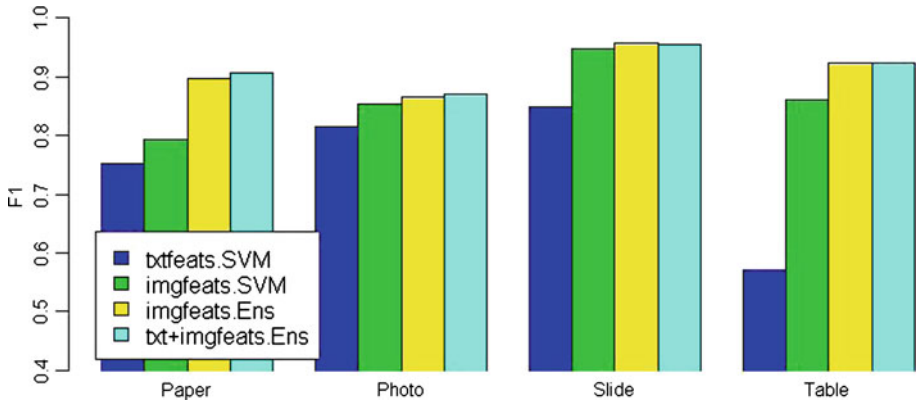

Fig. 5 $F1$-score page genre identification performance by genre for the methods: txtfeats.SVM, imgfeats.SVM, imgfeats.Ens, and txt+imgfeats.Ens

form the best plain SVM classifier (imgfeats.SVM) for all genres. As might be expected for the Table genre, the $F1$-score using the text-based method (txtfeats.SVM) is much lower than the methods that employ features that capture layout.

Comparing these results with Fleiss' kappa labeler agreement shown in Table 2, note that for the four genres, the rank order of the labeler kappa values is the same as the rank order of $F1$-score for the two methods using an ensemble classifier (imgfeats.Ens and txt+imgfeats.Ens). This indicates that page genre identification performance is better when labeler agreement is higher.

5.4 Discussion

Although Stamatatos et al. [31] had good results using text features, applying their text features to our data resulted in much poorer performance. We believe this is due in large part to the different sets of genres used. In [31], the corpus is a subset of Wall Street Journal documents composed of four genres: Editorials, Letters to the Editor, Reportage, and Spot News. Documents in these genres always contain words and are usually composed of complete sentences. Slides and tables tend to contain fewer words, and photos contain little or no words. With fewer stop words and punctuation characters, the features used in [31] may have been too sparse. Thus, supplementing them with selected text terms led to improved performance for our coarse genres.

It has been suggested that there are advantages to performing genre identification without text features. Scholl et al. [27] mention language independence and Kim and Ross [18] mention less language dependence and freeing the process from "text processing tools with encoding requirements and problems relating to special characters". They also note that "*pdftohtml* failed to extract information from seventeen percent of the documents."

Boese and Howe [2] examined whether different features for web genre prediction are transferable between web page corpora. They noted that *terms* in a web page are generally not transferable. They also noted that the only features common across the corpora they examined are a style readability measure, number of web links, and some HTML table tags. These features are independent of the terms in a document, as are our image-based features.

In agreement with earlier web page genre detection results by [20,27], our office document genre identification results indicate that using only text features is inferior to our image-based features. The latter are surface features capturing layout without requiring explicit segmentation into zones or classification into structural regions types. Our results also indicate that the addition of simple text features does not significantly influence performance for our four coarse genre types that exhibit differences in layout. Thus, we expect our image-based features with an ensemble classifier to be applicable to other collections of office documents. However, for finer grained genre classification where layout is similar, text features such as the stop words used by Stamatatos [31] and the surface cues proposed by Kessler et al. [16] could play an important role.

6 Application to document search

We applied our genre identification work to document search. Since document search is commonly *document*-based, rather than *page*-based, we first discuss labeling documents by genre. In particular, we describe how we use our office document data set for the document genre identification task and our experiments in predicting document genre based on the page genre classifications. We then describe a document browsing interface that makes use of the predicted genres as search facets.

6.1 Document genre data set

To develop a *document* genre identifier and evaluate performance, a set of documents where each *document* is labeled by genre is needed. Instead of performing a separate manual

labeling of documents, we assign a document to a particular genre if at least half of its pages have been manually tagged as that genre. The right column of Table 1 shows the number of documents in the corpus tagged with each of the four genres. A total of 798 tags were assigned to 1,178 documents. Note that there are documents without any tags.

6.2 Document genre identification

To automatically tag *documents* by genre, the genre scores for each *page* in a document are used. Since page classification is imperfect, we again employ learning and explore the utility of two feature representations for classifying a document based on the page scores. Since we biased the genre classifiers toward positively labeling a page as the specified genre, we assume that a negative page score does not contain much differentiating information; thus, we focused on the positive page scores. We investigated encoding the page scores in two different ways: (1) quantize the page scores in a document into a small number of bins and (2) combine the page scores produced by the classifiers for one genre into a single score.

The quantization method is illustrated in Fig. 6. Quantization of the page scores produced by the pair of classifiers is performed by providing two bins for negative scores and four bins for positive scores, with bin breakpoints at $\{-\infty, -1.0, 0.0, 0.33, 0.67, 1.0, +\infty\}$. To use the page scores as features, a feature vector is created that is composed of the counts of the quantized page scores for each of the four target genres, plus the number of pages in a document. Using Eq. 1, the feature vector values are scaled and shifted to range between 0 and 1 and then are used in an SVM.

In our second method, which is based on "score fusion", we combine a document's page scores for one genre into a single score (Fig. 7) and use it as a feature in an SVM. We again focus on the positive scores produced by the classifiers. Instead of making a hard decision as in Sect. 4.3, the maximum page score, $s_{\max}(p, g)$, produced by each pair of classifiers for page p and genre g is used in computing the summary score (Fig. 7b). To insure that the document genre score, $S_d(g)$, for document d and genre g is "normalized" to range between 0 and 1 and simultaneously focus on the positive scores in the predominant score range, each maximum page score $s_{\max}(p, g)$ is clipped to a maximum value of 1.0 and a minimum value of 0.0 (Fig. 7c). $S_d(g)$ is computed as the average of the individual maximum page scores for a document (Fig. 7d). Thus, $S_d(g)$ is computed from the pairs of scores as:

$$S_d(g) = \frac{1}{P} \sum_p \max(\min(s_{\max}(p, g), 1.0), 0.0)$$

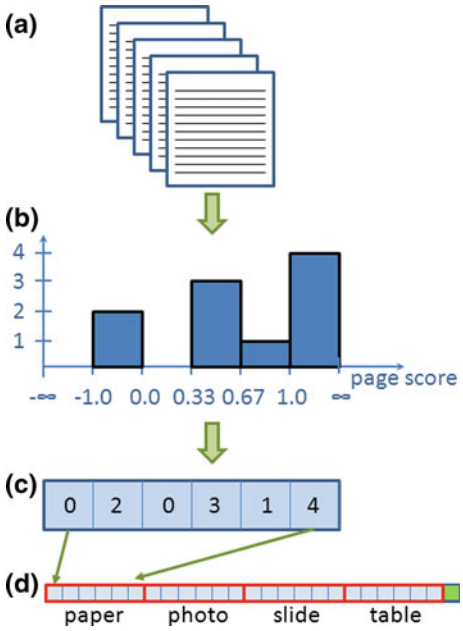

Fig. 6 Quantization of page scores to create a document feature vector. **a** For a given genre, each page in a document is scored by a pair of classifiers. **b** The scores are quantized and a histogram created. **c** Counts of the quantized scores for one genre. **d** Counts for each of the four genres (*blue*) and the number of pages (*green*) form the feature vector

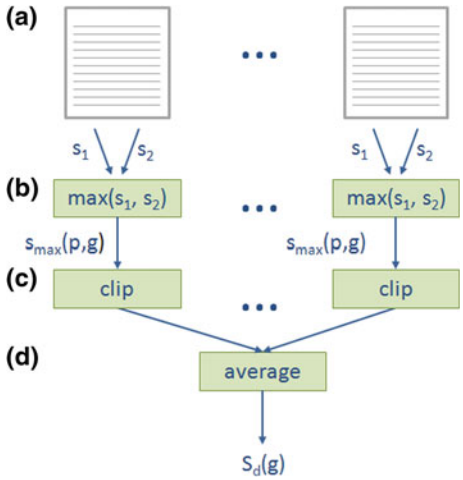

Fig. 7 Fusion of a document's page scores for one genre. **a** Each page of a document is scored by a pair of classifiers trained to identify one genre. **b** The maximum score of the pair is selected. **c** The maximum score is clipped. **d** A document's page scores are averaged, producing the document genre score, $S_d(g)$

where P is the number of pages in document d. $S_d(g)$ for each of the four genres plus the normalized number of pages in the document form the feature vector used in an SVM.

For each feature set, a separate classifier is optimized for each genre. The parameters of an SVM classifier are optimized using the same cross-validation approach as used for

page classification. The method using quantized features is referred to as **Quantized.SVM**, and the method based on score fusion is referred to as **Score.SVM**.

6.3 Document genre performance

In this section, we examine the performance of three methods for predicting document genre from page genre. Specifically, we compared the two methods described in the previous section, Quantized.SVM and Score.SVM, against a baseline voting classifier. Voting is a common classification method for combining the results from different classifiers on the same data. Here, we use voting to combine classification results on different data: the pages of a document. In particular, the page classification scores from the SVM classifier are used to vote whether to tag a document with a genre.

We expect that a trained classifier should better model the noisy features due to page genre identification errors, resulting in better performance than simple voting. Table 7 shows the results for the different document genre classification methods. The score fusion method, Score.SVM, had the best overall performance, with an $F1$-score of 0.8695.

The supervised methods exhibited statistically significant better $F1$-score performance than voting. Quantization was not better overall than our score fusion method. It may be that there is not enough data for good estimates in the larger number of dimensions, and that in the quantized representation, the features for documents with a small number of pages are noisy, hurting performance.

6.4 Genres in document search

Genre can play a valuable role when searching for office documents in a business organization. Unlike the web, where ranking of search results is strongly influenced by links between documents, business documents rarely contain links. Consequently, other methods for locating documents are needed. Faceted search has been used to locate documents based on their attributes (e.g., [11]), and automatically identified genres have been proposed as a useful facet for such searches [12]. To explore this, we applied our genre identification method to tag documents in a system we developed for faceted search and browsing of a collection of business documents. The system has been used internally to access 30,000 office documents and 2,30,000 images created over the past ten years.

Table 7 Comparison of document genre identification performance for different page combination models

Genre	Precision	Recall	$F1$-score
Voting	0.7120	$\mathbf{0.9524}^{s}$	0.8042
Quantized.SVM	0.7870^{v}	0.9316^{s}	0.8429^{v}
Score.SVM	$\mathbf{0.8474}^{vq}$	0.9027	$\mathbf{0.8695}^{vq}$

Superscripts 'v', 'q', and 's' indicate statistically significant improvement at the 0.05 level over voting, Quantized.SVM, and Score.SVM models, respectively. The best performance in each column is in bold

The system provides users with search and navigation options through metadata and the document collection file structure. Filters for different facets, including automatically computed genre, are provided by the interface. The results are displayed within the user-created directory hierarchy (Figs. 8, 9). Directories without matching documents are not displayed. Directories are visualized by three thumbnails of sample documents, cropped to squares to better fill the directory box. If fewer than three matching documents exist in a directory tree, then only that number of thumbnails is presented.

To provide genre tags for the search and browse system, we used the genre models trained on the office document data set described in Sect. 3 to tag the documents and images in the corpus. A user can restrict the genres shown in the interface by selecting one of the genres listed in the genre option as shown in Fig. 8. The pull-down menu includes 'Slide', 'Technical Paper', 'Photo', 'Table', and 'Other'. The 'Other' option contains any document that was not tagged with a genre. This option allows a user to use the genre facet to limit the documents presented to exclude the tagged genres. Examples of documents that fall into this category include memos, maps, and brochures.

Figure 8 displays the results of selecting the 'Tech paper' genre. Figure 9 shows the interface when the genre 'Slide' is selected. The background color of each directory is a different intensity of orange to indicate the strength of the match according to the match score. The darker intensities indicate that those directories have a higher proportion of documents tagged with the selected genre. Note in Figs. 8 and 9 that most of the thumbnails representing each directory are (cropped) technical papers and slides, respectively. The tool tip in Fig. 9 displays a larger image of the first slide from one of the presentations. One can see that a different set of directories is listed in Figs. 8 and 9 and that directories appearing in both listings, such as 'Activities' or 'Conferences', are represented by different document thumbnails.

As would be expected given imperfect genre identification performance (and the subjectivity of manual genre labeling), there are some genre tagging errors. However, the number of errors is small and the directories and documents presented are greatly reduced, enabling a user to more quickly browse for slides (or documents of another genre) in the directories.

The system has been demonstrated to many visitors who commented that the addition of genre as a search facet was very useful for narrowing the number of documents. They were excited by the directory thumbnail display that only

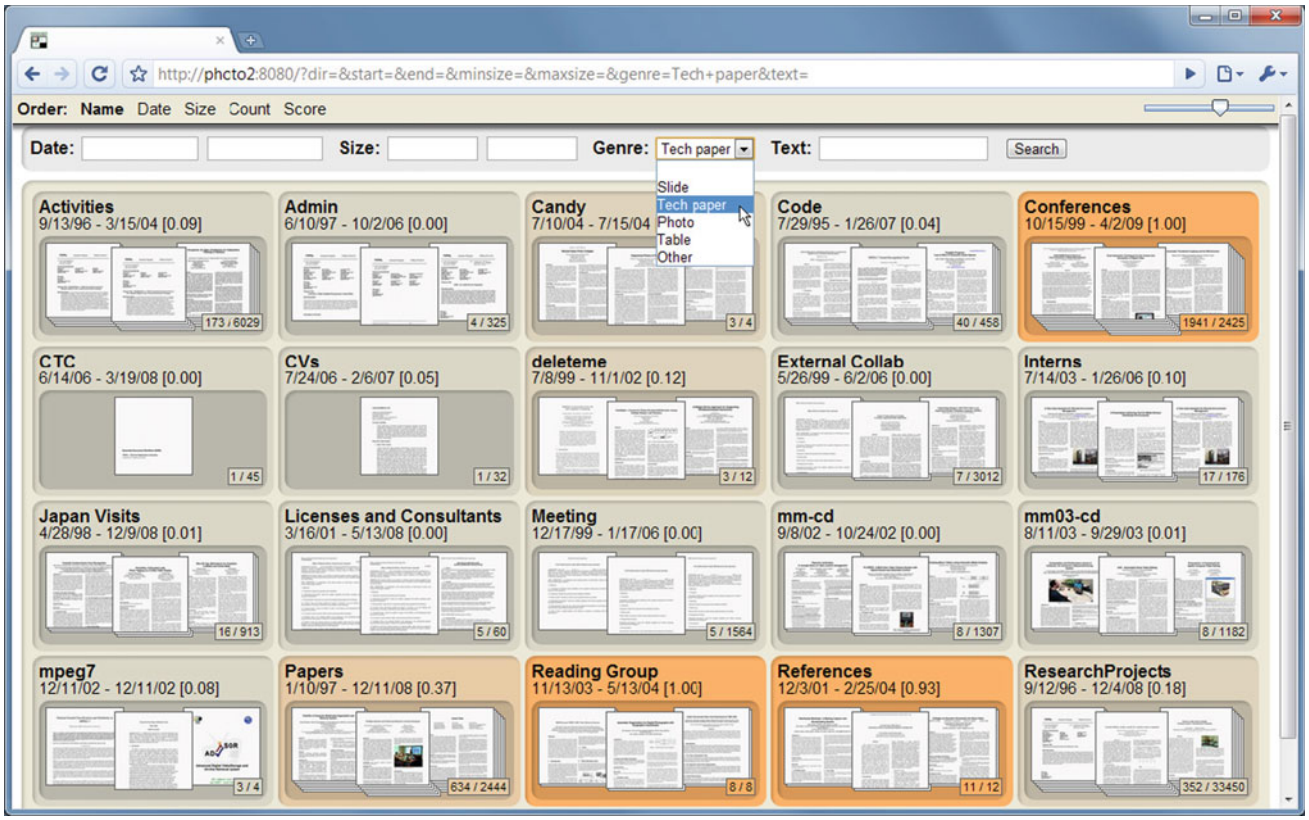

Fig. 8 Directories with the 'Tech paper' genre in an office document search and browsing system

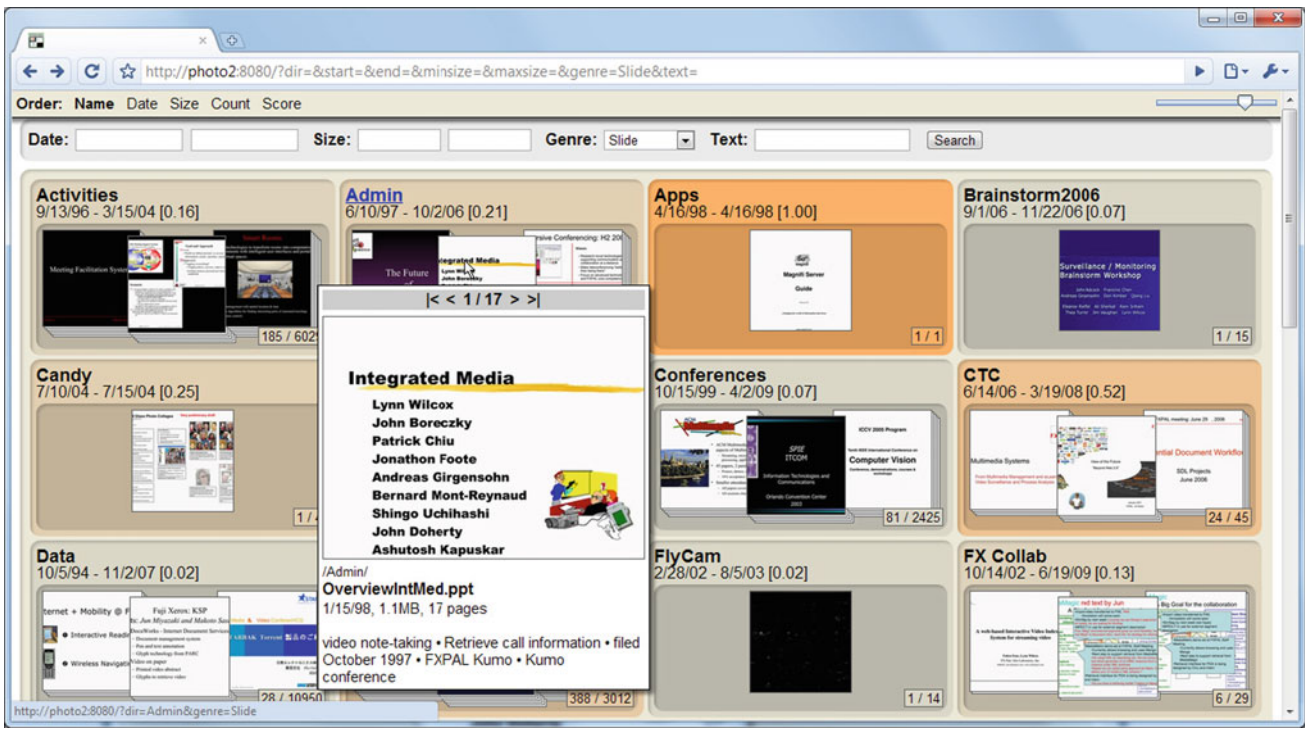

Fig. 9 Directories with the 'Slide' genre

showed documents of the selected genre. In one instance, we impressed a visitor by how quickly we could locate a slide from his previous visit ten years ago by using a combination of the slide genre and date facets. The system has also been used consistently by several people in our lab for finding old slides or photos to use when creating a new presentation.

7 Conclusions

We investigated design considerations for systems that perform coarse genre classification of office documents. Our set of document-inspired, visual, surface-type features do not require explicit classification into region types, unlike most earlier image-based systems. Experiments comparing these features against a simpler baseline set of image features showed that our visual features perform significantly better in identifying page genre as measured by precision, recall, and balanced $F1$-score. Feature selection also improved performance of a text-based genre classifier over a baseline, frequent terms-based classifier, indicating that enhancing the feature set with selected additional terms can improve performance over use of stop words only.

The results show that our image features perform significantly better than our text features. Additionally, the text features do not significantly improve performance in combination with image-based features. Thus, image-based features that capture the layout of a document can be more informative for coarse genre identification than simple textual features. As reported by others, performing genre identification without textual features enables broader applicability to new languages and documents, since textual features may vary with topic.

A simple ensemble method using data partitioning significantly improved the performance of individual one-versus-rest SVM classifiers for page genre identification. Each of our genre models learned the features of the different types of pages that occur in a genre, making creation of separate page-type identification models unnecessary.

We incorporated genre identification into a system for browsing and searching a collection of business documents, visibly reducing the document search space. We compared two methods for identifying document genre from page genre scores and used the best method to identify document genres in the collection.

In the future, we would like to investigate whether modeling the sequence of page genre labels improves document genre identification. We also would like to extend our methods to identify other genres that would be useful for office documents. And finally, we would like to conduct a user study to examine how genre affects search performance and perceived workload.

References

1. Bagdanov, A., Worring, M.: Fine-grained document genre classification using first order random graphs. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 79–83 (2001)
2. Boese, E.S., Howe, A.E.: Effects of web document evolution on genre classification. In: CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, New York, NY, USA, pp. 632–639 (2005)
3. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**(2), 121–167 (1998)
4. Chen, N., Blostein, D.: A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int. J. Doc. Anal. Recognit. **10**(1), 1–16 (2007)
5. Meyer zu Eissen, S., Stein, B.: Genre classification of web pages: user study and feasibility analysis. In: Biundo, S., Fruhwirth, T., Palm, G. (eds.) KI2004: Advances in Artificial Intelligence, pp. 256–269. Springer, Berlin (2004)
6. Fleiss, J.L.: Measuring nominal scale agreement among many raters. Psychol. Bull. **76**(5), 378–382 (1971)
7. Freund, L., Clarke, C.L.A., Toms, E.G.: Towards genre classification for IR in the workplace. In: IIiX: Proceedings of the 1st International Conference on Information Interaction in Context, pp. 30–36 (2006)
8. Gupta, M.D., Sarkar, P.: A shared parts model for document image recognition. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition, pp. 1163–1172 (2007)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explorations **11**(1), 10–18 (2009). http://www.cs.waikato.ac.nz/ml/weka/
10. Hao, X., Wang, J., Bieber, M., Ng, P.: A tool for classifying office documents. In: Proceedings of the Fifth International Conference on Tools with Artificial Intelligence, pp. 427–434 (1993)
11. Hearst, M.A.: Design recommendations for hierarchical faceted search interfaces. In: Broder, A.Z., Maarek, Y.S. (eds.) Proceedings of the SIGIR 2006 Workshop on Faceted Search, pp. 26–30 (2006)
12. Henderson, S.: Genre, task, topic and time: facets of personal digital document management. In: CHINZ '05: Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction, ACM, New York, NY, USA, pp. 75–82 (2005)
13. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification, (2010). http://www.csie.ntu.edu.tw/~cjlin/libsvm/
14. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: CVPR '97: Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition, pp. 762–768 (1997)
15. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: ECML '98: Proceedings of the 10th European Conference on Machine Learning, Springer, London, UK, pp. 137–142 (1998)
16. Kessler, B., Nunberg, G., Schütze, H.: Automatic detection of text genre. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 32–38 (1997)
17. Kim, Y., Ross, S.: Feature type analysis in automated genre classification (2007). http://eprints.erpanet.org/128/
18. Kim, Y., Ross, S.: Examining variations of prominent features in genre classification. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (2008)
19. Lee, Y.B., Myaeng, S.H.: Text genre classification with genre-revealing and subject-revealing features. In: SIGIR '02:

Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, pp. 145–150 (2002)
20. Levering, R., Cutler, M., Yu, L.: Using visual features for fine-grained genre classification of web pages. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (2008)
21. Manning, C., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, Chap. Text classification and naive bayes, Cambridge University Press, Cambridge (2008)
22. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. on Pattern Anal. Mach. Intell. **27**, 1226–1238 (2005). http://penglab.janelia.org/proj/mRMR/index.htm
23. Rauber, A., Müller-Kögler, A.: Integrating automatic genre analysis into digital libraries. In: Proceedings of the Joint Conference on Digital Libraries (2001)
24. Roussinov, D., Crowston, K., Nilan, M., Kwasnik, B., Cai, J., Liu, X.: Genre based navigation on the web. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences, vol. 4, IEEE Computer Society, Washington, DC, USA (2001)
25. Santini, M., Sharoff, S.: Web genre benchmark under construction. Special issue: automatic genre identification issues and prospects. J. Lang. Technol. Comput. Linguist. **25**(1):129–145 (2009)
26. Schölkopf, B., Burges, C., Smola, A. (eds.): Advances in Kernel Methods—Support Vector Learning, Chap. 11 Making large-scale SVM learning practical. MIT-Press, MA (1999)
27. Scholl, P., Domínguez García, R., Böhnstedt, D., Rensing, C., Steinmetz, R.: Towards language-independent web genre detection. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web, New York, NY, USA, pp. 1157–1158 (2009)
28. Shin, C., Doermann, D., Rosenfeld, A.: Classification of document pages using structure-based features. Int. J. Doc. Anal. Recognit. **3**(4), 232–247 (2001)
29. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2 (2003)
30. Snoek, C.G.M., Worring, M., Smeulders, A.W.M.: Early versus late fusion in semantic video analysis. In: MULTIMEDIA '05: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, pp. 399–402 (2005)
31. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Text genre detection using common word frequencies. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING2000), pp. 808–814 (2000)
32. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann, MA (2005)
33. Wong, K., Casey, R., Wahl, F.: Document analysis systems. IBM J. Res. Dev. **26**(6), 647–656 (1982)