DOCUMENT IMAGE SUMMARIZATION WITHOUT OCR

Dan S. Bloomberg and Francine R. Chen

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
E-mail: {bloomberg, fchen}.parc@xerox.com

ABSTRACT

A system for selecting excerpts directly from imaged text without performing optical character recognition is described. The images are segmented to find text regions, text lines and words, and sentence and paragraph boundaries are identified. A set of word equivalence classes is computed based on the rank blur hit-miss transform. This information is used to identify stop words and keywords. Sentences for presentation as part of a summary are then selected based on keywords and on the location of the sentences.

1. INTRODUCTION

There exist no efficient techniques that use analysis and generation of natural language for creating computer-generated summaries of ascii text. Instead, practical summarization methods use selection and extraction of sentences [8, 5, 6, 9]. Summaries of imaged documents have been created, using such text-based methods, by applying optical character recognition (OCR) to the images as a preprocessing step to generate an ascii text representation. However, OCR is computationally expensive, visually error-prone and, for applications where a quick summary is appropriate, it may be unnecessary.

In contrast to these text-based techniques, we propose a method for automatically creating a summary from an imaged document without recognition of the characters in each word. Our system for summarization does not rely on language analysis or generation, and in this respect follows work by others on ascii text [6, 5]. In this approach, a few short passages, or excerpts, are extracted from an imaged document for presentation as a "summary" of the document. Additionally, imaged documents have more information, such as font size, placement of text, and included images, than is available in plain text documents. Some of this information can be used to identify headers and create a table of contents for the document. Only text regions composed of the *dominant* font size, for which enough samples are present, are considered for sentence extraction. Sentences are selected for extraction based on frequency statistics of their words.

Figure 1 outlines the steps in performing text image summarization. In the next two sections, we describe the image processing and summary image selection techniques used to create a summary. Word images are grouped into equivalence classes based on shape similarity, which can be performed much more quickly than OCR. Stop words are identified based on statistical characteristics of the word equivalence classes in each document. The location of sentence and paragraph boundaries are used, along with statistical information on the words, to generate summary scores for each sentence.

2. IMAGE PROCESSING

To identify a set of summarizing excerpts, both layout and logical information is extracted from the document. Image segmentation is performed in a top-down fashion, and the primary steps are shown in Figure 2. Each image is deskewed and all halftones and other "image" parts are removed [1], leaving text and line graphics. Larger groupings, such as textblocks and graphic blocks are then coalesced, using a morphological closing operation to join components within such regions. In this process, care must be taken to avoid joining adjacent columns. This is done by subtracting from the closed image a mask that is made from the vertical whitespace by inverting the original image and opening with a large vertical structuring element.

Dominant font textblocks and words

To separate textblocks from graphics regions, morphological operations are performed to identify textlines. For each region, a horizontal morphological closing is used to join characters in the underlying image, solidifying any textlines that may exist, and then the statistics of the resulting components are analyzed. The key scaling factors are the median width and height. If the width-to-height ratio is sufficiently large, and if the median width is a significant fraction of the region width, then the region is assumed to be a textblock.

Textblocks are classified into two sets, depending on how

0-7803-3258-X/96/$5.00

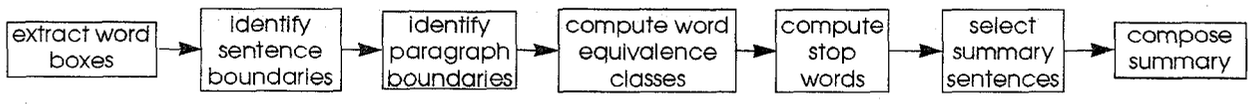

Figure 1: Text image summarization system.

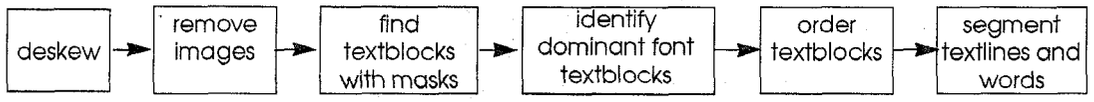

Figure 2: Segmentation for summarization system.

close the median font size and line spacing are to the median of the entire document. Textblocks whose median size and spacing are within about 15 percent of the document medians are deemed to conform to the dominant font; others are segregated. Headers and other important information are typically found in the latter set, with relatively large font sizes.

The conforming textblocks are next analyzed for reading order. The general difficulty is that top-to-bottom and left-to-right compete for priority in a complicated and non-standardized way. We model the competition both by using a hierarchical top-to-bottom decomposition and by distinguishing between regions that have either horizontal, vertical or no overlap. Details can be found elsewhere [3].

The textlines are located using a morphological closing with a horizontal structuring element that is sufficiently large to connect all parts of each textline into a single connected component. Then each of the textline images is segmented into words, again using a horizontal morphological closing to merge characters within each word. This is done conservatively, and is followed by a second merging step using bounding boxes of the resulting components [3].

Sentences

Sentences and paragraphs are found from text in the dominant font. Sentences are identified by searching for periods near the baseline of the textlines, and finding the words most closely associated with the periods. We use a set of tests based on measured distances. It is important to use a scale for these comparisons that is based on the size of the font being examined, and is independent of the resolution at which the image is scanned. We choose this scaling parameter to be the measured median height of the bounding boxes of connected components for the characters in the textblock. This is typically the "x-height" of the predominant font.

Identification of periods is somewhat tricky, because it is necessary to distinguish a period that ends a sentence from noise pixels near the baseline, commas and semicolons, a dot in an ellipsis, the lower dot in a colon, and a dot that ends an intra-sentence abbreviation. We must include periods that are part of question and exclamation marks. And for full sentence identification, it is necessary to include punctuation, such as quotes and parentheses that may follow a period, as part of the sentence.

Components are identified that are "period-shaped" and within 4 pixels vertically of the computed baseline. The condition for being period shaped is that neither the maximum width nor height exceeds 0.4 (of the x-height) and the difference between the height and width does not exceed 0.12 (of the x-height). If the dot passes these tests, we check if it forms part of an ellipsis, colon, exclamation or question mark by examining nearby component shapes and locations.

The final test is the most difficult: to differentiate between an intra-sentence abbreviation and a period. We first check if the left edge of the following component is farther than 0.4 (of the x-height) from the right edge of the component. Otherwise, to determine if the next component starts a new sentence, check its height. If it does not descend below the baseline and extends above the baseline by a distance approximating the maximum distance for characters on the textline, then it has the shape and location of a capital letter. This final test will not miss any true periods, but it will mis-classify some intra-sentence abbreviations as periods. Once the periods have been located, each word whose bounding box has a right side closest to a period is tagged as a sentence-ending word.

Paragraphs

There are two primary ways in which paragraphs can be laid out: indentation of the first line and extra inter-line spacing. We presently use indentation and ignore the latter, because inter-line spacing is rarely used for paragraphs but is often used typographically to set aside special text, such as equations. Nevertheless, our method will find most new paragraphs even if indentation is not used.

Paragraphs are identified at the granularity of sentences. A sentence starts a new paragraph if (1) it starts a new textline and either (2a) the previous textline is not flush right with the textline before it, or (2b) the new textline is indented to the right with respect to the previous textline. In the event that

the new textline starts a new text column, (2b) is amended to check indentation with respect to the following textline rather than the previous one. This method fails in the rare situation where indentation is not used with paragraphs and the previous paragraph ends flush right, so that the only indication of a new paragraph is the extra inter-line spacing. We do not use other layout information, such as the location of section headers between textblocks, to reduce the number of missed paragraph beginnings.

At this point the segmentation phase is complete. All the words in the predominant font have now been put into reading order and labeled by their location in the document, as well as the sentence and paragraph to which they belong.

Equivalence classes

An unsupervised classifier is used to place each word in the dominant font into one of a set of equivalence classes. The classifier compares word images using either a rank blur hit-miss transform [2] or a similar rank version of the Hausdorff metric [7]. Rank versions of pattern matchings are required to allow some outlier pixels in the word matches. This prevents a small amount of random image noise from invalidating the match and putting instances of the same word into different classes. In these rank comparisons, the maximum number of outlier pixels allowed is taken to be a fraction of the word image area. From the number of instances in each word equivalence class, word frequencies can be estimated.

It is important that classes are not often split, and to that end a small number of words that are assigned to the wrong class can be tolerated. To reduce class splitting, each word can be analyzed for terminal punctuation, such as a comma, period or hyphen. If a comma or period is attached to the word, it should be removed for the matching process. If a word terminates in a hyphen, it can be assigned to a special class and ignored in later statistical analysis.

A particular difficulty with this approach is that the optimum parameters (blur size, fraction of allowable outlier pixels) are different for matching small and large words. Suppose that a fraction of the pixel outliers, relative to the image area, are permitted in order to compensate for image variations and misalignment between images of the same word. If the outlier fraction is made sufficiently large to avoid splitting classes for short words, then the number of outlier pixels allowed for long words may permit matching of some words that are otherwise well-aligned but differ by a character. Thus, we optimally should reduce the fraction of outlier pixels allowed for larger words. Another way to achieve this effect is to perform a second matching, using a larger structuring element for the blur and a much tighter threshold on the number of outliers. This second match has little effect on the matching of small words, but acts to prevent matching images of large words that are different. The two matches must both be valid for a word image to be placed in an existing class.

In the classification process, each word in the document is successively analyzed for a match with the representative of an existing class. If a match is found, it is added to the list of instances for that class; otherwise, a new class is formed with the word image as the class representative.

One pass of the unsupervised classifier is typically sufficient. For each word encountered, the *best* match against an existing class representative is used. A greedy algorithm is more efficient, but the best match gives better results and is preferred because both unsupervised classification procedures are relatively fast.

A number of steps can be taken to increase the efficiency of the classification process. First, all class representatives are sorted by size, and matches are attempted to a subset differing in width and height by a small amount (typically 3 pixels for 150 ppi images). Second, to optimize matching speed, word-aligned dilated images of each word and class representative are pre-computed, matches are evaluated between these subimages, and matches are aborted when the accumulated number of misses exceeds the allowable outlier fraction in either direction [2]. Third, the number of word classes can be capped at a maximum value, beyond which any words not matching an existing template are placed in a class that has no weight in later scoring. With this maximum, the classification time increases linearly with the document size; otherwise, the number of classes has an asymptotic growth that is linear in the document size. Capping the number of word classes is a reasonable constraint for this application, where most of the important words can be expected to occur early in the document.

From here on, the text analysis can be performed on these tokens, with no reference back to the image except for image composition for output display.

3. SUMMARY EXCERPT SELECTION

Summary sentences are extracted using algorithms based only on statistical characterizations of words in a document, without regard to possible meanings of the words. In contrast to some of the text-based summarization approaches described in [6, 9], which require auxiliary corpus information, only the information within a document is used to construct a summary. This is crucial in image summarization based on the use of equivalence classes, both for speed and because words in different documents are typically in different fonts, preventing simple decisions across documents based on comparing bitmaps.

Stop words

Words that are not content words (commonly called *stop* words) are first identified. Word frequencies, word locations, and word image widths are used to rank equivalence classes

as to their likelihood of being a stop word. Generally, a word is more likely to be a stop word if it has high frequency, small width, and is rarely the first word in a sentence. The list of stop words is chosen by selecting the N highest ranking words, where N is dependent on document length and other characteristics of the document [4]. Then, a small set of high frequency words that are not stop words are chosen as *keywords*. All other non-stop words are considered to be *content* words.

Summary sentences

Each sentence is scored based on a number of features: (1) the number of keywords contained in the sentence and the number of times each of these keywords occurs in the document, (2) the location of the sentence in the document, and (3) the location of the sentence within the containing paragraph. Sentences that contain at least one keyword are referred to as "thematic" sentences. The feature scores are derived from training data and represent the probability of a particular feature occurring conditioned on the sentence being a summary sentence [5]. Each of the three criteria is treated as independent, and the final score of each sentence is generated from the individual scores, in this case by taking their product.

The number of sentences to be selected for a summary is specified by the user. Then, the set of sentence excerpts can be composed to form one or more summary images. Informal evaluation of the selected summary sentences indicates that this image-based method produces "indicative" types of summaries [8].

- The Common Gateway Interface (CGI) and the <FORM> tag in HTML were created to give us some control over their interaction with the Web.
- By creating applications that use the <FORM> tag and CGI, developers can allow users to do simple things such as enter search criteria and strings, control navigation by making selections from lists, or even play games.
- Most CGI applications (commonly referred to as scripts) are simple UNIX filters created from shell scripts, Perl, or Awk.
- More complex applications are actual programs written in a compiled language such as C or C++.
- Therefore, you need to be careful about tasks such as opening files for write, connecting to a database, or breaking other operations if multiple applications attempt to access a resource concurrently.

Figure 3: Five summary sentences.

4. SAMPLE RESULTS

A sample excerpt of five sentences from a six-page imaged document is shown in Figure 3. The lines of text corresponding to the selected sentences are extracted from the imaged document and have not been reformatted to a standard line width. A bullet precedes each selected sentence to encourage the reader to think of each sentence as a separate highlight. The selected sentences are presented in order in which they occur in the document, rather than by score, for better readability. The first sentence has no keywords, but was selected primarily because of its location. The last four sentences are thematic, and additionally, they all either begin or end a paragraph.

5. REFERENCES

[1] D. S. Bloomberg, "Multiresolution morphological analysis of document images," *SPIE Conf. 1818, Visual Communications and Image Processing '92*, Boston, MA, Nov 18-20, 1992, pp. 648-662.

[2] D.S. Bloomberg and L. Vincent, "Blur Hit-Miss transform and its use in document image pattern detection," *SPIE Conf. 2422, Document Recognition II*, San Jose, CA, Feb 6-7, 1995, pp. 278-292.

[3] D.S. Bloomberg and F.R. Chen, "Extraction of text-related features for condensing image documents," *SPIE Conf. 2660, Document Recognition III*, San Jose, CA, Jan 29-30, 1996, pp. 72-88.

[4] F.R. Chen and D.S. Bloomberg, "Extraction of thematically relevant text from images," *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, April 15-17, 1996, pp. 163-178.

[5] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, pp. 68-73, 1995.

[6] H.P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, pp. 159-165, 1959.

[7] G. Matheron, *Random Sets and Integral Geometry*, J. Wiley and Sons, NY, 1975.

[8] C. Paice, "Constructing literature abstracts by computer: Techniques and prospects," *Information Processing and Management*, vol. 26, pp. 171-186, 1990.

[9] L.C. Tong, and S.L. Tan, "A statistical approach to automatic text extraction," *Asian Library Journal*, **3(1)**, pp. 46-54, Mar 1993.