Algorithms **2012**, *5*, 490-505; doi:10.3390/a5040490

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

The Effects of Tabular-Based Content Extraction on Patent Document Clustering

Denise R. Koessler [1], Benjamin W. Martin [1], Bruce E. Kiefer [2] and Michael W. Berry [1,*]

[1] EECS Department, Min H. Kao Building Suite 401, University of Tennessee, 1520 Middle Drive, Knoxville, TN 37996, USA; E-Mails: dkoessle@eecs.utk.edu; bmarti15@eecs.utk.edu

[2] Catalyst Repository Systems, 1860 Blake Street, 7th Floor, Denver, CO 80202, USA; E-Mail: bkiefer@catalystsecure.com

* Author to whom correspondence should be addressed; E-Mail: berry@eecs.utk.edu; Tel.: +1-865-974-3838; Fax: +1-865-974-5483.

Received: 1 July 2012; in revised form: 16 August 2012 / Accepted: 9 October 2012 / Published: 22 October 2012

Abstract: Data can be represented in many different ways within a particular document or set of documents. Hence, attempts to automatically process the relationships between documents or determine the relevance of certain document objects can be problematic. In this study, we have developed software to automatically catalog objects contained in HTML files for patents granted by the United States Patent and Trademark Office (USPTO). Once these objects are recognized, the software creates metadata that assigns a data type to each document object. Such metadata can be easily processed and analyzed for subsequent text mining tasks. Specifically, document similarity and clustering techniques were applied to a subset of the USPTO document collection. Although our preliminary results demonstrate that tables and numerical data do not provide quantifiable value to a document's content, the stage for future work in measuring the importance of document objects within a large corpus has been set.

Keywords: text mining; patent documents; table data

---

1. Introduction

Automated document understanding is commonly used to improve web searches and document content indexing [1]. However, information and data within a document can be represented in a variety

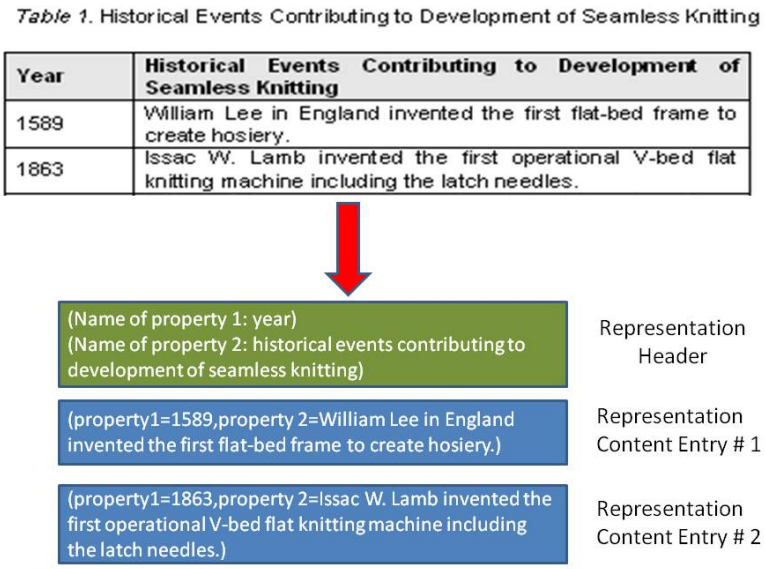
of forms such as a paragraph of text, a table, or an image. Further, one document can contain multiple tables, images, and paragraphs of text, and the same document can contain different representations of the same content. On a much larger scale, a set of documents may contain similar information; however, their data representations may be very different. With so many ways to represent information, the automated task of document content extraction and representation becomes problematic.

Additionally, differences in document layout make it difficult to automatically extract relationships within the data. In the same corpus, opposing layouts complicate the ability to detect relationships between different documents. Moreover, conflicting document layouts make it difficult to aggregate relevant data within the same document. However, the use of standard data formats greatly facilitates finding and organizing the most relevant data in a document [2].

This work is part of an ongoing collaboration between the University of Tennessee and Oak Ridge National Laboratory to study the use of visual gaze tracking to analyze document layout. The collaboration aims to apply gaze tracking analysis to note which document objects a person observes and the order in which these objects are observed. A *document object* is defined to be any element of a document such as a paragraph, image, table, *etc.* The data collected from the gaze tracking analysis helps to identify which objects in the document are of the most importance as well as the relationships among the objects. An object's importance can then be determined by analyzing the amount of time that a person spends looking at an object while data relationships can be determined by noting the order in which the person views objects in the document, respectively.

Another collaborative goal is to aggregate document content into objects. The generated collection of objects will establish a new set of metadata that describes each document, such as the aggregation shown in Figure 1. Each object in the metadata will contain a *tag* describing what the object is and how it is related to other objects in the document. Once the metadata is created, it can be used to identify documents that contain similar data. Further, the metadata can assist with searching within the document set and generating summaries of the data contained within the corpus.

Figure 1. Metadata extraction.



With the above goals in mind, we have recently created ground truth metadata for United States patents. After rigorous preprocessing using techniques of data exploration and text mining, we have created an XML data format to describe content found in the patent documentation. We restricted our study of document clustering to documents that contain table objects in order to assess the value (with respect to content) of tabulated data in the associated HTML files.

Our procedures were implemented on document sets with and without numerical words. By *numerical word* we refer to any element in the patent document which contains at least one number. The words *1988*, *1a*, and *010111January* are examples of numerical words. The inclusion or exclusion of numerical words from a document set's dictionary is applied to determine if such words contribute a measurable value to a document's content. A methodology to evaluate which indexing model best clusters subsets of patent documents was also developed.

2. Related Work

The marriage of machine learning and document processing is of particular interest in the areas of web search, electronic discovery, and document content indexing [3–8]. In addition, much work has been done in standardizing the modeling of document content. Some of these methods attempt to model the structure of each document and then cluster documents with similar layouts [9]. Other research considers the logical structure of a document by learning the reading order for the objects in a document [10].

Previous case studies have explored the use of text mining techniques on patent document processing. As critiqued in [11], early approaches examined and extracted document relationships solely from a patent's citations. Following these citation analysis techniques, the research presented in [6,8] examines the strengths and weaknesses of linguistic tools for automated patent document understanding. Further, the work presented in [7] applies feature selection, term association, and cluster generation to extracting topics and phrases from patent documents.

The theoretical background for vector space indexing and similarity models used in this work is fully described in [12]. Additional foundational theory on the use of inverse document frequency for document processing is found in [13] and [14]. The documentation and algorithms implemented in the MATLAB-based Text to Matrix Generator (TMG) software used in our work can be found in [15].

3. Input Data

The corpus used in this study comprised approximately 2.5 million public-domain files describing patents granted by the United States government. Patents were archived as a combination of TIFF images, HTML documents, and PDF documents. One of the challenges when working with such a corpus is the fact that some of the documents are stored as plain TIFF images which are not easily parsed into document objects. In this study, the TIFF images were preprocessed but not analyzed for content.

Another challenge is that the PDF files are merely converted versions of the original HTML files. Hence, the PDF data files simply contain a copy of the information that can be more easily read from the HTML files. In addition, many of these PDFs were not converted properly from HTML and thus were blank. As with the TIFF images, the PDF files of corpus were only preprocessed.

Unfortunately, the HTML files were not without problems as well. They were created using Optical Character Recognition (OCR) which caused embedded tables in the HTML to become jumbled and unstructured collections of data. Deriving document object relationships from such output proved to be quite challenging. The final corpus used in this study comprised 1.5 million HTML files, of which approximately 13% contained tables.

4. Objectives

The main focus of this work is to produce metadata that describes the content for a subset of U.S. Patent HTML documents and to determine the importance of the table data in the documents. This metadata consists of identifiers for the objects contained in the documents as well as information on the content contained in the objects. Such metadata can be used as ground truth data for future text analysis of U.S. Patent corpora.

In order to facilitate the collection of this ground truth metadata, we have exploited the Perl programming language [16] and a text mining module developed in MATLAB [17] to automatically extract, tag, and quantify objects within a document. Preprocessing millions of documents for object extraction was fairly simple using Perl's regular expression matching capability. We use the MATLAB-based Text to Matrix Generator (TMG) [15] for document similarity calculations based on derived term-by-document matrices.

A few of the important tasks needed to accomplish our objectives are briefly summarized below:

4.1. HTML Content Analysis

The HTML patents seem to be very similar in structure, but the data in the HTML patents is obscured by HTML markup. This task concentrates on analyzing the HTML files, identifying meaningful portions of the data, and converting them to a more concise XML format. In addition to simply converting the patent data to XML, information describing the number of figures, tables and equations in each document is generated.

4.2. Table Identification and Extraction

Approximately 13% of the HTML patents contain tables that have been mangled due to OCR conversion. However, it is determined to be remarkably difficult, if not impossible, to reconstruct these tables due to the degree of irregularity in the data introduced by OCR. Therefore, we extract the table data from the patents into separate files. This task, of course, requires effective ways of recognizing and extracting tables with a maximal degree of accuracy.

4.3. Document Clustering by Similarity

The TMG software environment creates a term-by-document matrix from a given input directory. An appropriate MATLAB function can then be used to calculate the cosine similarity between two document vectors. Given a cosine similarity, we aim to determine how the similarity between two documents differs

with and without the tabular data. To do so, we must cluster the document similarity values with and without the HTML tables.

4.4. Evaluation Strategy

A testing strategy must be capable of answering the following research questions:

1. Do tables in HTML patents provide quantifiable content to a document when applying a *bag of words* model?
2. Do *numerical words* in patent documents affect document similarity?
3. Which indexing model best clusters the patent corpus? Similarly, what weighting and frequency constraints on dictionary words achieve the best clustering results?

5. Methodology

In this section, we described the formal procedures used to accomplish the data/text mining tasks outlined in Section 4.

5.1. Irrelevant File Identification

The first step in preprocessing the data of the patent corpus was the identification of files that do not contain relevant data. It was determined that the following types of files would not be useful in any sort of analysis:

1. HTML files that do not contain full text for their respective patents,
2. PDF files that do not contain full text for their respective patents, or do not contain any text at all, and
3. duplicate TIFF files.

Identification of files that do not contain full text was the simplest of these tasks. A simple text search for the string "Full text is not available for this patent" determined if a file does not contain full text. If this string was present in either a HTML or PDF file, the file was irrelevant.

Identifying blank PDF files posed a more difficult problem. After careful analysis of the blank PDF files, it was determined that the blank PDF files do not have a font defined anywhere in the file. Therefore, searching the PDF files for the text `\Font\` was the easiest way to find these blank files. Any PDF files not containing this string was considered irrelevant.

For the identification of duplicate TIFF files, the complicating factor was that the data in the files was stored as binary image data. This made it difficult to identify any distinguishing aspects of the TIFF data which could indicate that a file is a duplicate. To address this issue, we used the MD5 checksum of each TIFF file and stored the checksums in a Perl hash. If it was determined that two TIFF files have the same MD5 checksum, the files were declared as duplicates and one of them was marked as such.

5.2. HTML Content Analysis

Once the irrelevant files were identified and removed, we found that, contrary to our assumption, the HTML files were also difficult to parse. While HTML has a very regular structure, each file contained an unexpectedly large amount of formatting markup. In addition, there were many cases of missing closing tags or stray characters interspersed throughout the HTML markup. When scaling up to thousands of documents, the irregularities and extraneous tags in the markup made it exceptionally difficult to process the HTML files.

In order to facilitate parsing, an effort was made to extract the patent data from the HTML files and store it in a simpler XML format. This was accomplished by taking advantage of the fact that the individual HTML files for the patents were fairly uniform in structure. Thus, using Perl-based regular expressions sufficed to extract the data and store it in the new XML format.

In addition to the XML format, the patent data in the HTML files was analyzed to determine the number of *objects* (figures, tables and equations) in each patent. Counts of the number of objects removed from each patent were also recorded. In order to determine the number of figures and tables in a document, the number of unique tables and figures referenced were counted for a given patent. Once the number of tables and figures was determined, the totals for all of these objects were included as part of the summary XML file for the patent. If the same object was referenced more than once, the first reference was counted toward the summary total and all subsequent references to that object were ignored.

Careful examination of the patent files revealed that many of the objects were replaced with specialized placeholders. Table 1 provides an example of some of these placeholders. An official list of all such placeholder types was not available at the time of this study.

Table 1. Common Placeholders in HTML Patents.

Object Type	Example Placeholder
Equation	##EQN01##
Table	##TBL01##
String	##STR01##
Special	##SPC01##

While these placeholders may be of some use in determining the number of objects stored in a patent, the usage of these placeholders was by no means consistent. For example, not all tables were replaced with TBL placeholders. In many cases, the tables were converted to text via OCR and included in the patent data. However, it was also common for the tables to be excluded and replaced with SPC placeholders instead of TBL placeholders. In light of this, all TBL placeholders were ignored. In addition, the STR placeholders had a tendency to become mangled when a large number of strings were excluded from a patent. This caused the STR placeholder text to extend into the patent text. Because of this, these STR placeholders were also ignored. However, the EQN placeholders were very consistent and were used to count the number of equations in each patent.

5.3. Table Identification and Extraction

In order to determine the relevance of the table data for the patent corpus, a Perl script was developed (all preprocessing scripts are available by request at http://cisml.utk.edu/.) to manipulate the table data in the HTML patent files. The first feature added to this tool was the ability to identify HTML files that contain tables. This script was tested on a subset of approximately 49,000 files to determine the percentage of files that contain tables.

Once these files were identified, their structure was analyzed to determine ways to identify tables in the HTML data. Analysis of the structure of the HTML patents revealed that most tables were stored in a single line of a file and that these lines usually begin with the text `<BR><BR>Table`. A Perl script was created to search a HTML file for lines beginning with this text and extract these lines from the file. Once these lines were extracted, three new files were created: A file that contained all of the data in the patent but with the HTML tags removed (referred to hereafter as the *Clean* patent document subset), a second file that contained all of the patent data except for the tables and HTML tags (referred to as *Tables Removed*), and a third file that contained only the table data (referred to as *Tables Only*). These files were later used to help determine the relevance of table data for representing patent information.

There were a few cases in which tables were not delineated by `<BR><BR>Table`. Exceptions were added to the script in order to handle these outlier cases. In addition, once data analysis began, it became apparent that numbers and words containing numbers were not interpretable. In light of this, another feature was added to the table extraction script that removed all word tokens containing numerical characters.

5.4. Indexing Parameters

The creation of a vector space model for the patent corpus was achieved using the MATLAB-based Text to Matrix Generator (TMG) software environment [15]. TMG utilizes a graphical user interface to implement MATLAB and Perl scripts for document indexing. Specifically, we used the TMG functions for reading the patent documents, creating a term dictionary, and generating a term-by-document matrix for subsequent document similarity calculations. Further, this software environment employs the popular *Bag of Words* model for its indexing scheme. In this model, individual words are considered without evaluating the order in which they appear in a document. This approach was selected for the case study as a means to gathering ground truth information regarding the data set of interest.

The TMG interface has many built-in parameters that directly affect the quantity and quality of the resulting term-by-document matrix. The filtering techniques that were extensively tested throughout this study included the local minimum and maximum term frequency thresholds, global minimum and maximum document frequency thresholds, local and global weighting, and the use of a stop words list (*i.e.*, removal of unimportant words such as *a*, *the*, and *of*).

The local thresholds limited the inclusion of a term in the dictionary based on the term's frequency within one document. For example, a local maximum value of 10 would exclude any word from the document dictionary that occurred more than 10 times within a document; whereas a local minimum value of 10 would exclude any word that occurred less than 10 times in a document. On the other hand, the global document frequency thresholds limited the inclusion of a term in the dictionary based on the

term's frequency throughout the corpus. That is, a global maximum threshold of 10 excluded a word from the document dictionary that occurred in more than 10 documents; whereas a global minimum of 10 excluded a term if it was found in fewer than 10 documents.

In addition to these thresholds, TMG permits multiple local and global term weighting strategies. In this study, we used two types of local weighting methods: Binary and term frequency. A *binary* local weight used a 0 or 1 to represent the presence or absence, respectively, of a term within a particular document. A *term frequency* local weight included the number of occurrences to represent a term's presence within a document, and 0 otherwise. On the global level, we used either no global scheme or the *inverse document frequency* (IDF) option. Equation 1 defines the inverse document frequency of term t_i with respect to document n_i in a set of N documents.

$$IDF\left(t_i\right) = log\left(\frac{N}{n_i}\right) \tag{1}$$

Using the parameters described above, three different models were implemented. Each model contained a unique combination of the local and global term frequency methods. Table 2 summarizes the three models used.

Table 2. The three models used for document indexing analysis.

Model Name	Local Weight	Global Weight
Term Frequency	Term Frequency	None
Binary IDF	Binary	IDF
Term Frequency IDF	Term Frequency	IDF

We utilized the SMART stoplist provided by Cornell University [18]. Common words unique to the patent corpus (e.g., *united*, *states*, and *patent*) were also added to the stoplist. Further, a second list of stop words was created for documents that included numerical words. This list contained common numerical information such as years, figure labels, and reference numbers.

While there are many numerical words which appear to be the result of inaccuracies with the OCR conversion (e.g., 1998Watson, 010111Jan), it is not guaranteed that all numerical words will be removed by term frequency. For example, the object label *1a* occurs multiple times within one document, and exists in almost every document. To ensure obvious unimportant words were not included in a documents term dictionary, additions were made to a stop list to include labels, years, dates, and other common numerical terms. While both the revised stop list and term frequency models will filter the occurrence of certain numerical words, there commonly exist certain measurements or statistics which describe the pertinent findings of the patent. Typically, the important measurements and/or the new results presented by a patent are repeated multiple times within a patent. Therefore, the frequencies of numerical quantities distinguish one patent from another. An additional objective was to quantify how these numerical words distinguished similar documents across an entire corpus of patents. As a result, this case study aims to explore how various term frequency weighting schemes affect the clustering of documents with and without numerical words.

5.5. Document Clustering by Similarity

Using appropriate TMG functions, a directory of documents was read into MATLAB and a corresponding term-by-document matrix was created. In the matrix, each element specifies the frequency of a term (row) in a particular document (column). A document-to-document similarity matrix was then generated using the cosine (see Equation 2) of any pair of document vectors d_i (columns) of the term-by-document matrix.

$$cosine\,(i,j) = \frac{d_i \cdot d_j}{\|d_i\| \cdot \|d_j\|} \tag{2}$$

Document clustering was achieved using the agglomerative clustering module (based on average Euclidean distance similarity scores) provided in MATLAB. Since MATLAB limits the resulting dendrogram to no more than 30 clusters, scatter plots were also generated so that one could simply visualize the cluster assignments of the patent documents.

5.6. Evaluation Strategy

Two methods were used to analyze the effectiveness of the methodologies described in Section 5.5. First, given two document-to-document similarity matrices, the percent difference between each score was calculated. The minimum, maximum and average percent differences were recorded. Additionally, the standard deviation of the percent differences was calculated. The evaluation of the dendrograms and clustering, on the other hand, was more subjective. To illustrate the effectiveness of this approach, we include information regarding the structure of the dendrograms and the clusters via scatter plots.

To determine the best content representation of the patent corpus, we ran each of the three models shown in Table 2 and observed the percent differences between the *Clean* and *Tables Removed* patent document subsets. These trials were completed with a small subset of patent documents comprising 100 HTML files with tables and the exact same HTML files without tables.

Next, an experiment was run that quantified the value of the *Table Object* to the document's indexing score. Using the best observed model, we analyzed the dendrograms and clustering results for patent documents with and without HTML tables. This was completed using the small subset of data at first and then repeated using a much larger subset of the patent documents.

6. Results

Of the approximately 2.5 million files in the USPTO corpus, approximately 6% (or 158,290) of them were determined to be irrelevant or unusable. In most cases, these were either HTML or PDF files without full text or duplicate TIFF files.

Figure 2 illustrates a sample of the HTML markup for a single patent and the resulting XML markup that was generated for the same HTML data. Aside from the obvious benefit of simplifying the markup in order to make the HTML patents easier to process, one unexpected benefit of the conversion from HTML to XML was that the size of the patent information was significantly reduced. Specifically, this lead to a 14 GB reduction (or 21.2%) in the size of the patent information originally stored in the HTML files.

Figure 2. HTML content, on the left, and the resulting XML content, on the right.

```
<TABLE WIDTH="100%"> <TR><TD VALIGN="TOP" ALIGN="LEFT" WIDTH="10%">
Inventors: </TD><TD ALIGN="LEFT" WIDTH="90%">
 <B>Knight; Francis J.</B> (Piscataway, NJ)<B>, Konopka; Barbara A.</B>
 (Branchburg Township, Somerville, NJ) </TD></TR><TR> <TD VALIGN="TOP"
 ALIGN="LEFT" WIDTH="10%">Assignee:</TD>
<TD ALIGN="LEFT" WIDTH="90%">
<B>Johnson & Johnson</B>
(<BR></TD>
</TR><TR><TD VALIGN="TOP" ALIGN="LEFT" WIDTH="10%" NOWRAP>Appl. No.:
</TD><TD ALIGN="LEFT" WIDTH="90%">
<B> 04/732,410</B></TD></TR>
  <TR><TD VALIGN="TOP" ALIGN="LEFT" WIDTH="10%">Filed:
  </TD><TD ALIGN="LEFT" WIDTH="90%">
  <B>May 27, 1968</B></TD></TR>
</TABLE><HR><p><TABLE WIDTH="100%">
  <TR><TD VALIGN=TOP ALIGN="LEFT" WIDTH="40%"><B>Current U.S. Class:
  </B></TD>  <TD VALIGN=TOP ALIGN="RIGHT" WIDTH="80%"><B>
  604/232</B></TD></TR><TR><TD VALIGN=TOP ALIGN="LEFT" WIDTH="40%">
  <B>Field of Search: </B></TD>
  <TD ALIGN="RIGHT" VALIGN="TOP" WIDTH="80%">
128/218,218P,218NV,218.1,218.1P,215
  </TD></TR>
</TABLE>
```

```
<inventors>
<inventor>
<inventorName>Knight; Francis J.</inventorName>
<inventorCity>Piscataway</inventorCity>
<inventorLoc>NJ</inventorLoc>
</inventor>
<inventor>
<inventorName>Konopka; Barbara A.</inventorName>
<inventorCity>Branchburg Township, Somerville</inventorCity>
<inventorLoc>NJ</inventorLoc>
</inventor>
</inventors>
<assignee>
<assigneeName>Johnson &amp;amp; Johnson</assigneeName>
</assignee>
<appNo>04/732,410</appNo>
<fileDate>May 27, 1968</fileDate>
<USClass>604/232</USClass>
<fieldOfSearch>128/218,218P,218NV,218.1,218.1P,215</fieldOfSearch>
```

Extracting tables from the viable patent documents confirmed our previous observation that approximately 13% of the files contained tables. Table 3 shows the number of files for each patent directory parsed along with the corresponding number of files that were determined to have tables.

Table 3. Three directories of patent files on which the three models from Table 2 were applied. Each of the three subsets contained approximately 2100 files with tables.

Directory	Total File Count	Files with Tables
0	16,386	2155
1	16,382	2148
2	16,302	2157

Figure 3 illustrates the effect of the local minimum and maximum thresholds on the resulting percent size of the dictionary. The Y-axis displays the percent size of the dictionary in logarithmic scale, base 10. Consequently, the upper graph of Figure 3 illustrates that as the Maximum Local Term Frequency varies from 0 to 25, the size of the resulting dictionary ranges from approximately 75.8% to 100%. Similarly, the lower graph of Figure 3 shows that as the Minimum Local Term Frequency varies from 0 to 20, the size of the resulting dictionary ranges from approximately 100% to 5.6%. Although this effect is depicted for a smaller subset of the patent corpus, the trend was consistent for larger subsets of patent documents. We note that the local maximum threshold removed fewer words from the document set's dictionary than the local minimum threshold did. For the extreme value of the local maximum, the dictionary was limited to approximately 75% of its total capacity, whereas the extreme local minimum threshold eliminated all words from the document set's dictionary.

It is interesting to note some of the words conserved when extreme values of the thresholds were applied. When testing on a set of patent documents with numerical words and local maximum frequency threshold set to 1, typical dictionary words include: *2545866march*, *1988watson*, *a2*, *a3* and *a4*. Table 4 displays the final threshold levels implemented throughout all our remaining experiments. For this case study, the authors selected these thresholds in conjunction with the list of stop words to simultaneously

filter uninteresting words and include essential words in a document's dictionary. The precise filtering of the dictionary paired with a *Bag of Words* model enables the authors to establish a ground truth level of understanding of this corpus' similarity from a pure text mining perspective.

Figure 3. The effects of the percent size of the dictionary as the local thresholds change. The Y-axis represents the percent size of the dictionary in logarithmic scale, base 10.

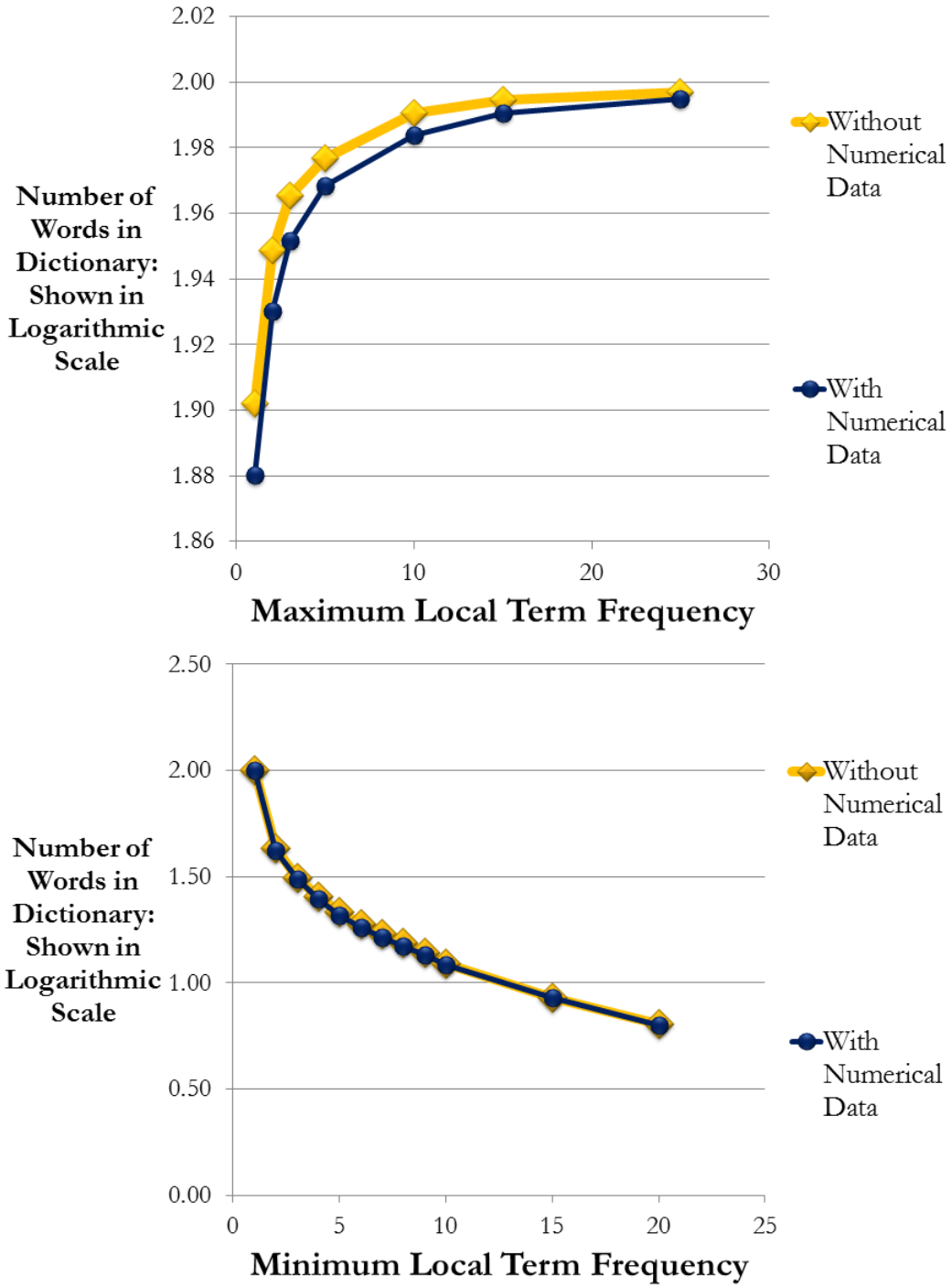


Table 4. The parameters selected and implemented for the main experiments; N is the total number of patent documents parsed.

Parameter Type	**Threshold**
Local Maximum	∞
Local Minimum	2
Global Maximum	$N-1$
Global Minimum	2

Table 5 shows the results for two experiments using the models from Table 2. Both experiments calculated the document-to-document similarity matrices using two instances of the same collection of patent documents: one with HTML tables and one without HTML tables. Experiment 1 examined the similarity matrices for the set of documents without numerical words and Experiment 2 used the exact same set of documents but included numerical words.

Table 5. Change in document-to-document similarity matrices when tabular information is excluded from patent documents; Experiment 1 removes numerical words and Experiment 2 includes numerical words.

Experiment 1	Min % Change	Max % Change	Average % Change	Standard Deviation
Term Frequency	0.0	81.8	4.1	6.8
Binary IDF	9.0×10^{-5}	74.7	5.3	7.2
Term Frequency IDF	5.0×10^{-5}	93.7	6.1	8.3
Experiment 2	**Min % Change**	**Max % Change**	**Average % Change**	**Standard Deviation**
Term Frequency	0.0	81.8	4.1	6.8
Binary IDF	2.0×10^{-2}	74.9	5.5	7.6
Term Frequency IDF	5.0×10^{-5}	93.7	5.9	8.3

Table 5 provides statistical evidence that supports our claim that numerical words within patent documents do not yield substantial content for document representation. As summarized in Table 6, minute differences evidenced in the results of Experiments 1 and 2 demonstrate that including numerical words does not improve document-based similarities. Further, the average percent change (in document-based similarities) between the document set with HTML tables and the document set without HTML tables ranged between 4% and 6%. Based on the percent change in similarity, Tables 5 and 6 confirm that the Term Frequency model is unaffected by the inclusion of tabular data or numerical words.

Table 6. Percent difference shown between models with numerical words, and models without numerical words.

Model	Average % Difference	% Difference in Standard Deviation
Term Frequency	0.0	0.0
Binary IDF	3.7	5.4
Term Frequency IDF	3.3	0.0

Figure 4 displays two dendrograms created by agglomerative hierarchical clustering of document-to-document similarities using the Binary IDF Model for patent document representation. The top dendrogram illustrates the clustering for patent documents without HTML tables and the bottom dendrogram shows the clustering assignments for the files with HTML tables. In both cases, numerical words were not included. To better visualize individual document-to-cluster assignments as well as any differences, Figure 5 provides a scatter plot of the same results in Figure 4. For two different models, Figure 5 plots the document number along with the bin in which the document was assigned. The bin assignments were determined by each document's average Euclidean distance similarity score. As illustrated in Figure 5, the documents were clustered into the same bin when the models were run with and without the tabular data. Hence, tabular data does not appear to have any effect a patent's similarity score.

Figure 4. Agglomerative hierarchical clustering results for the Binary IDF Model when numerical words are not included; tables are removed in the top dendrogram and are included in the bottom dendrogram.

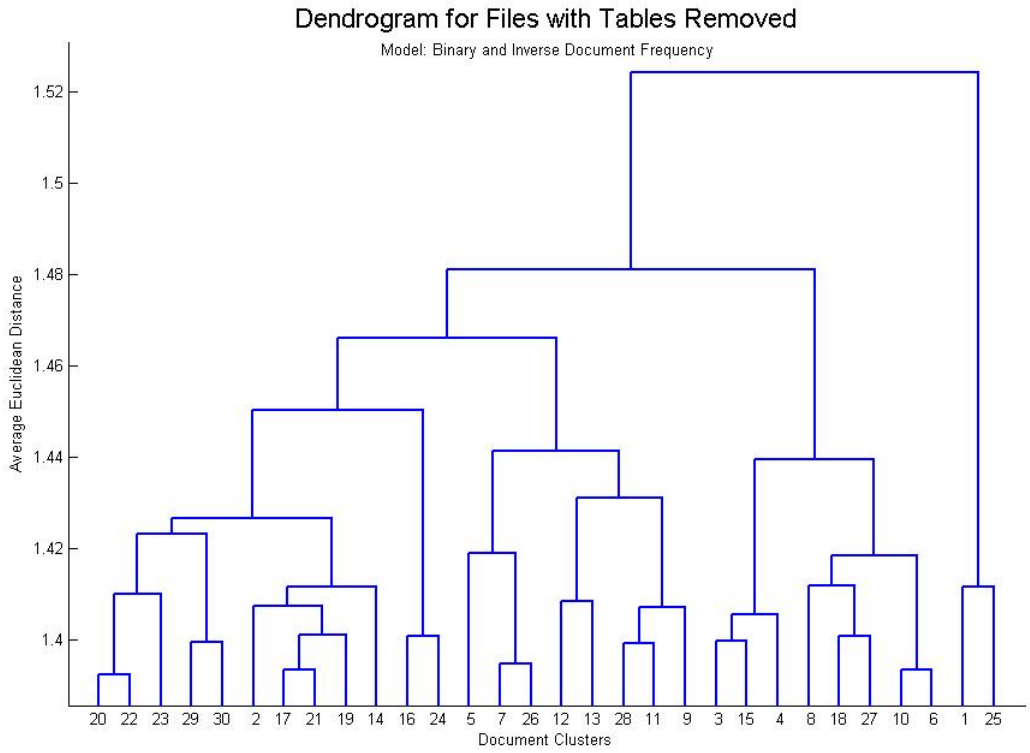


Figure 4. *Cont.*

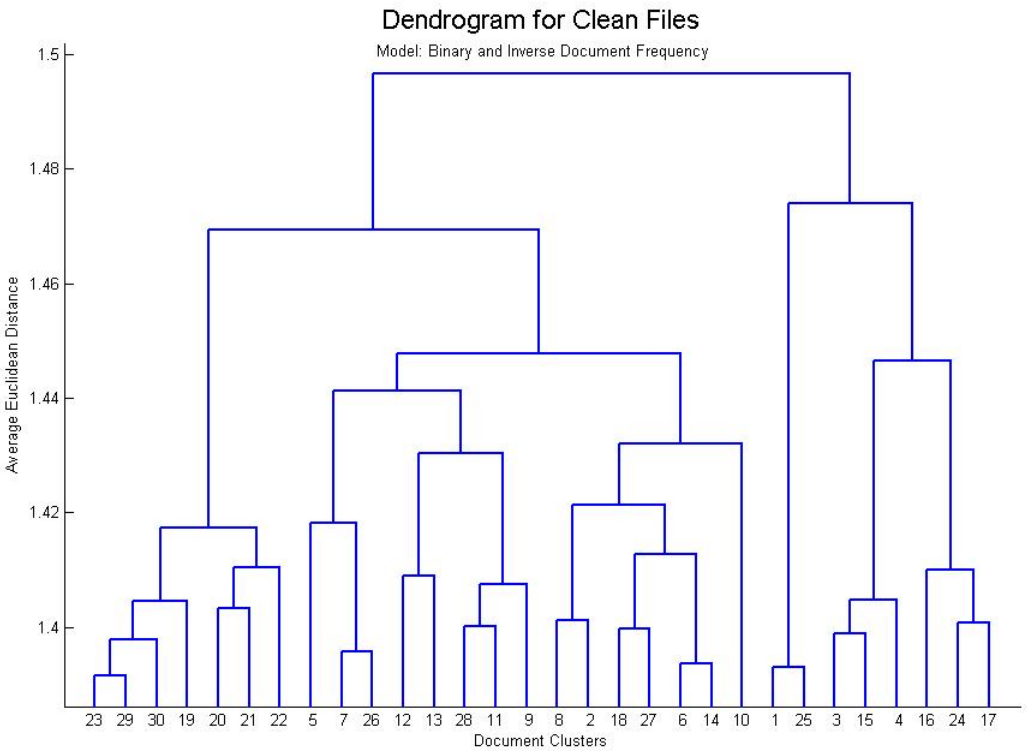
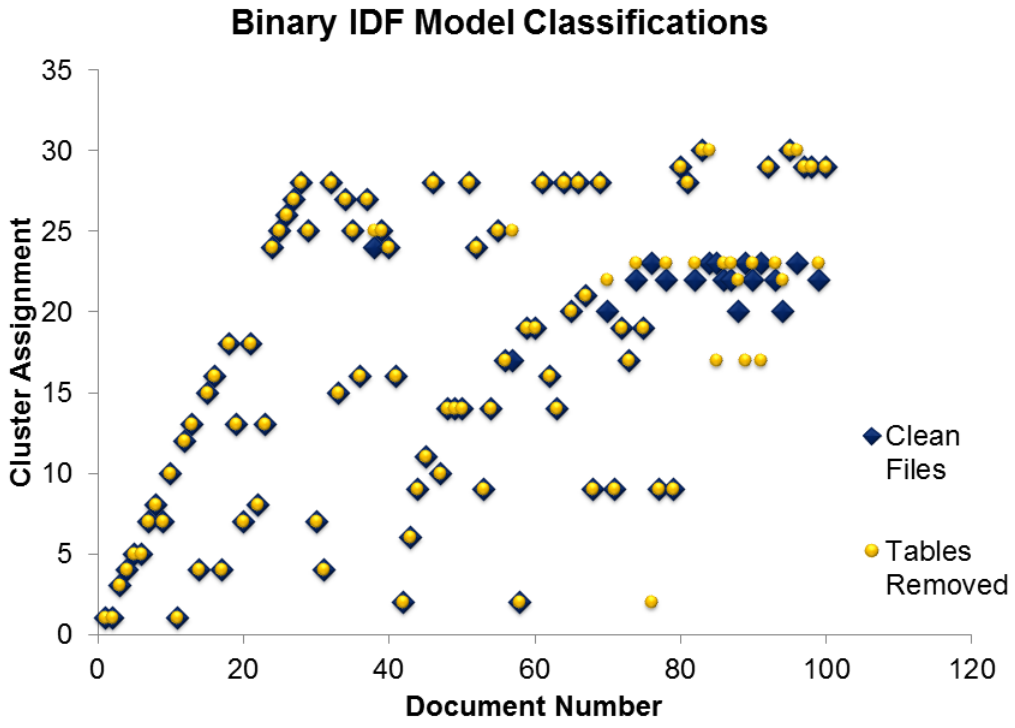


Figure 5. Scatter plot showing cluster assignments of Figure 4 by document number.



Hierarchical trees (dendrograms) and associated scatter plots were generated for every experiment in our study. The results for the other indexing models (Term Frequency and Term Frequency IDF) were

very similar and are not shown here. No significant differences in document clustering were observed with the inclusion of numerical words either.

7. Summary

We began this study by evaluating potential ways to parse the data in the USPTO documents. As part of this evaluation, we were able to develop software that converts the patent data stored as HTML to a simpler XML format. We also evaluated ways to generate a *bag of words* representation of the patent data for use in document analysis.

After the analysis of numerous parsing strategies and their resulting dictionaries, we have determined that most of the numerical words in the dictionary of any patent document subset were not actually part of the original document. These numerical words were primarily conversion mistakes for the date, year, and author of the patent document or clearly represented figure and table labels. Our findings in Section 6 certainly support the conclusion that the numerical words found in an HTML patent generally do not provide useful content for indexing the document.

Further, we have shown that table objects in the HTML files representing USPTO documents do not provide quantifiable content to the documents when applying a *bag of words* indexing model. In addition, we have demonstrated that numerical words do not affect document-to-document similarity. When considering the average percent difference of the document-document similarity matrices, the simple term frequency model is the most insensitive to the deletion of table objects. However, the similarity scores created by the Binary IDF model tended to yield the optimal document clustering.

References

1. Kochi, T.; Saitoh, T. A layout-free method for extracting elements from document images. In *Document Analysis Systems: Theory and Practice*; Lee, S., Nakano, Y., Eds.; Springer Berlin Heidelberg: Berlin, Germany, 1999; Volume 1655, pp. 215–224.
2. Gupta, G.; Niranjan, S.; Shrivastava, A.; Sinha, R. Document layout analysis and classification and its application in OCR. In *EDOCW '06. 10th IEEE International*, Proceedings of Enterprise Distributed Object Computing Conference Workshops, Hong Kong, China, 16–20 October, 2006; pp. 58–58.
3. Sharpe, M.; Ahmed, N.; Sutcliffe, G. An intelligent document understanding & reproduction system. *MVA* **1994**, 267–271.
4. Berry, M.; Esau, R.; Kiefer, B. The use of text mining techniques in electronic discovery for legal matters. In *Next Generation Search Engines: Advanced Models for Information Retrieval*; Jouis, C., Biskri, I., Eds.; IGI Global: Hershey, PA, USA, 2012; pp. 174–190.
5. Vincent, L. Google book search: document understanding on a massive scale. In *Proceedings of International Conference on Document Analysis and Recognition*; IEEE Computer Society: Los Alamitos, CA, USA, 2007; Volume 2, pp. 819–823.
6. Yoon, B.; Park, Y. A text-mining-based patent network: Analytical tool for high-technology trend. *J. High Technol. Manag. Res.* **2004**, *15*, 37–50.

7. Tseng, Y.H.; Lin, C.J.; Lin, Y.I. Text mining techniques for patent analysis. *Inf. Process. & Manag.* **2007**, *43*, 1216–1247.
8. Fattori, M.; Pedrazzi, G.; Turra, R. Text mining applied to patent mapping: A practical business case. *World Pat. Inf.* **2003**, *25*, 335–342.
9. Farrow, G.S.D.; Xydeas, C.S.; Oakley, J.P.; Khorabi, A.; Prelcic, N.G. A comparison of system architectures for intelligent document understanding. *Signal Process.: Image Commun.* **1996**, *9*, 1–19.
10. Malerba, D.; Ceci, M.; Berardi, M. Machine learning for reading order detection in document image understanding. In *Machine Learning in Document Analysis and Recognition*; Springer: Berlin, Germany, 2008; pp. 45–70.
11. Michel, J.; Bettels, B. Patent citation analysis. A closer look at the basic input data from patent search reports. *Scientometrics* **2001**, *51*, 185–201.
12. Berry, M.; Drmac, Z.; Jessup, E. Matrices, vector spaces, and information retrieval. *SIAM Review* **1999**, *41*, 335–362.
13. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520.
14. Papineni, K. Why inverse document frequency? In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2001, pp. 25–32.
15. Zeimpekis, D.; Gallopoulos, E. TMG: A MATLAB toolbox for generating term-document matrices from text collections. In *Grouping Multidimensional Data: Recent Advances in Clustering*; Springer: Berlin, Germany, 2006, pp. 187–210.
16. The Perl Programming Language Home Page. Available online: http://www.perl.org/ (accessed on 16 October 2012).
17. MathWorks Home Page. Available online: http://www.mathworks.cn/ (accessed on 16 October 2012).
18. Cornell SMART Project English Stoplist. Available online: ftp://ftp.cs.cornell.edu/pub/smart/english.stop (accessed on 16 October 2012).