The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence
Scholarly Big Data: AI Perspectives, Challenges, and Ideas:
Technical Report WS-16-13

Automatic Summary Generation for Scientific Data Charts

Rabah A. Al-Zaidy
alzaidy@psu.edu
The Pennsylvania State University
University Park, PA

Sagnik Ray Choudhury
sagnik@psu.edu
The Pennsylvania State University
University Park, PA

C. Lee Giles
giles@ist.psu.edu
The Pennsylvania State University
University Park, PA

Abstract

Scientific charts in the web, whether as images or embedded in digital documents, contain valuable information that is not fully available to information retrieval tools. The information used to describe these charts is typically extracted from the image metadata rather than the information the graphic was initially designed to express. The problem of understanding digital charts found in scholarly documents, and inferring useful textual information from their graphical components is the focus of this study. We present an approach to automatically read the chart data, specifically bar charts, and provide the user with a textual summary of the chart. The proposed method follows a knowledge discovery approach that relies on a versatile graph representation of the chart. This representation is derived from analyzing a chart's original data values, from which useful features are extracted. The data features are in turn used to construct a semantic-graph. To generate a summary, the semantic-graph of the chart is mapped to appropriately crafted protoforms, which are constructs based on fuzzy logic. We verify the effectiveness of our framework by conducting experiments on bar charts extracted from over $1,000$ PDF documents. Our preliminary results show that, under certain assumptions, 83% of the produced summaries provide plausible descriptions of the bar charts.

Introduction

Scientific charts contain valuable information for search engines. The metadata used to tag the chart image does not always provide enough information about the graphical contents of the chart. This has led many studies to address the problem of automating chart reading and understanding. Typically, the chart will represent data values that could be expressed as tables in a database. For a user, the values become known by visualizing the chart. However, for a computer to read the chart, image processing techniques must be involved. Additionally, the designer of the chart may intend to illustrate a certain fact that the data values support, and this illustration is achieved by means of graphical effects in the chart. The purpose of this study is to propose a framework that is able to infer this fact from both the data values of the chart and the graphical effects employed in it's design by producing a corrosponding summary in plain text. We focus on a specific class of charts, bar charts.

Summary generation from numerical data is a natural language problem. Linguistic studies propose constructs called protoforms that facilitate various forms of sentence generation. Protoforms generate linguistic summaries from data by exploiting the structure of the input data. Thus, in order to apply this method, the chart data is required to have some semantic structure. We propose to structure the chart data as a semantic graph. The appeal of a semantic-graph structure for the chart is due to it's wide applicability and ability to provide an abstract of the chart data. This abstraction of the chart data can be easily employed in various knowledge extraction and information retrieval applications. We provide an empirical evaluation of the proposed framework on a set of 300 bar charts extracted from scholarly PDF documents. The evaluation shows the results of key steps in the pipeline and the effects they propagate to subsequent steps. An overall evaluation of the summaries is conducted to verify the effectiveness of the system in automating the reading of bar charts.

Related Work

Various studies provide approaches to address the problem of understanding scientific graphics, algorithms, and tables in scholarly documents (Liu et al. 2007), (Kataria et al. 2008), (Lu et al. 2009), (Tuarob et al. 2013), and (Fang et al. 2012). Earlier work that focused on classification and data extraction of scientific charts in particular are (Huang and Tan 2007) and (Yang, Huang, and Tan 2006). More recently, (Chen, Cafarella, and Adar 2015) proposes a search engine called *DiagramFlyer* that indexes a large amount of scientific charts extracted from PDF documents. The method extracts the text from the charts and classifies their role, i.e. x-axis, y-axis, and uses it along with the figure metadata. Their method, described also in (Chen, Cafarella, and Adar 2011), does not show any module to extract the original data contained in the graphical components of the chart. Another approach for searching and retrieving charts is proposed by (Li et al. 2015), which identifies certain structures that are extracted from charts and user-entered queries. A proposed linear model is then used to rank the charts whose information structures are best matched to those in the queries. One

of the extracted structures is the notion of an *intended message*, which they describe in (Elzer, Carberry, and Zukerman 2011). The method is based on the idea that a chart contains communicative signals which are extracted and used in a Bayesian network-based method to infer the intention of the bar chart. A further extension of this work is used to aid the visually impaired in reading graphical charts by means of an interactive chart summarizing tool called *SIGHT*, described in (Demir et al. 2010).

Any chart summarizing approach builds on a data extraction method that uses image processing techniques to extract the data values of the chart. The extraction method in (Al-Zaidy and Giles 2015) uses connected component analysis of the chart image. This is based on the method proposed by (Savva et al. 2011). Other methods as in (Kataria et al. 2008), target plot charts and apply machine learning techniques to extract the data. Some methods are based on an analysis of gray level histograms, such as the one proposed by (Chester and Elzer 2005).

The problem of generating linguistic summaries from charts has been addressed by many projects, such as the iGRAPH-Lite project (Ferres et al. 2007) that presents the data facts contained in the chart as a textual summary. Another summarization system is the one proposed by (Demir, Carberry, and McCoy 2008), where they provide the intended message, described above, as a summary. Summarizing numerical data and time series is also the focus of various studies. An example is the method for generating summaries from data collected by medical-purpose sensors by (Wilbik, Keller, and Alexander 2011). The method generates the summary of eldercare data using protoforms, a concept proposed by (Zadeh 2002).

Method

The process of generating a summary for a chart follows the pipeline shown in Figure 1. The first step is to extract the original data values from the bar chart. The next step is to extract the graphical effects that may contribute to the summary. Once the features are extracted the chart components are labeled and stored in a semantic graph. The summary generation module is responsible for reading the graph and applying the linguistic techniques to generate the summary. The sections below provide a description for each module in the pipeline.

Bar Chart feature Extraction

The first step towards understanding a chart's content is to extract the original data values from the graphical component. The method described in (Al-Zaidy and Giles 2015) is used to extract the following data values from the bar chart: (1) x and y axes names, if they exist, (2) numerical value for each bar (the y value), (3) nominal value for each bar (name of the data value the bar represents). Once the data is recovered from the chart image, it is then passed to the feature extraction method.

The graphical features are special graphical effects used by the designer of the chart to illustrate a certain fact. Based on empirical analysis of charts messages performed in (Elzer, Carberry, and Zukerman 2011), the most common messages charts typically aim to convey, are: increasing/decreasing trend, rank of a specific value or subset of values, maximum/minimum value among a set of values. In this work we select messages that are more prominent among bar charts. Each of these messages is associated with certain graphical features in the chart. Arranging the bars in increasing or decreasing order can be considered an indicator that the chart is illustrating a trend. A bar that has a different color than all other bars in a single-series chart is an indicator that the bar may be of specific importance to the meaning of the chart. Other annotations, such as labeling only certain bars with their value or other textual data suggests the designer would like to draw attention to their values. In (Elzer, Carberry, and Zukerman 2011), they refer to these as *Salient* bars. If none of the bars in the chart image have any salient elements, nor display a visible trend, it is most likely that the purpose of the chart is to simply display the values and how they rank to one another. In (Elzer, Carberry, and Zukerman 2011) the message of such a chart is referred to as communicating the bars' *Rank*.

Based on the findings mentioned above, we select three type of features to extract from the chart data: Salience, Trend, and Rank. Bars are marked Salient if they have a distinct color or are annotated with texts. If the bar values are monotonically increasing, or decreasing, this indicates the presence of the Trend feature. Additionally, if the x-axis labels are a time-series or have ordinal values, this is a feature labeling the data series as a whole. As for the Rank feature, we provide the user with only the maximum and minimum values found in the chart for simplicity. Once the extraction of the data and the features is complete, the next step is to generate a semantic-graph to represent the data. This process is described in the next section.

Semantic-Graph Representation of Bar Charts

Constructing a semantic graph requires definition of relevant semantics. Semantic labels are derived from the roles of each extracted data value or feature. The y-axis name has semantic label *element*. If the y-axis name contains a noun and a descriptor, the descriptor is labeled *attribute*. The x-axis name has semantic label *parameter*. The edge between the attribute (or Element of no attribute was found) is labeled with the features extracted by analyzing the data. Three labels are used for edges. *hasTrend* can be: increasing, decreasing, or none. *hasSalient* determines whether any of the bars are salient, either by annotated text or a different color. *hasMaxMin* specifies the maximum value of the bars and the minimum . This is used to to describe the chart in case no other features are present.

Summary Generation

To generate the summary we follow a method similar to (Wilbik, Keller, and Alexander 2011) which is an adaptation of the concepts initially proposed in (Zadeh 2002). The method is based on generating linguistic summaries from protoforms. A protoform is a linguistic construct that allows the use of structured data to generate English language sentences using fuzzy logic. We define 4 basic protoforms

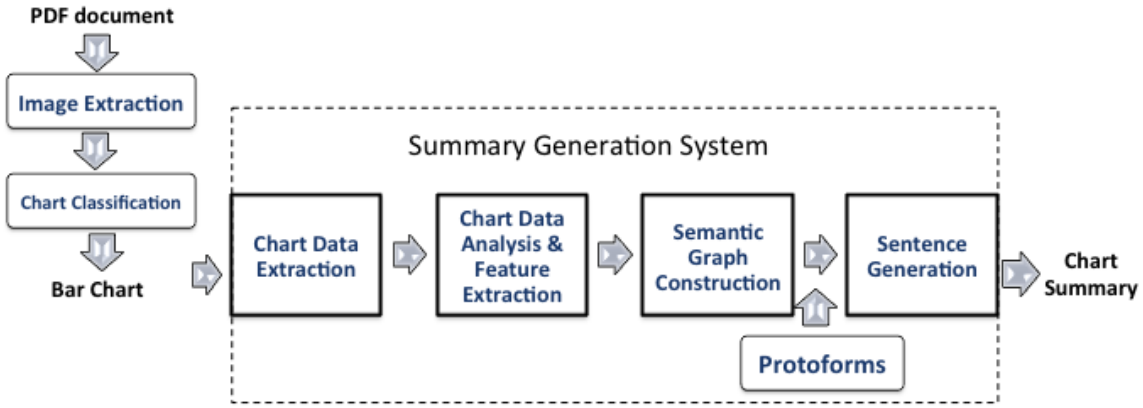

Figure 1: System Architecture

Algorithm 1: Algorithm to Generate Summary from a Chart's Semantic-Graph

Input : Semantic Graph G
Output: Summary Sentence S

if $hasTrend$ *is not* nil **then**
  $S =$ protoform 1 for time series
  $S =$ protoform 2 for ordinal series.
  $S = nil$ otherwise.
end
if $hasSalient$ = *TRUE* **then**
  $S = S+$ protoform 3
end
if $hasSalient$ = *FALSE and* $hasTrend$ = nil **then**
  $S =$ protoform 4
end
return : S

for each type of feature the relationship between the *element* node and *parameter* node may share. The protoforms are: trend protoform for a time series, trend protoform for an ordinal data set, salience protoform, and max/min protoform. We use notation similar to that in (Wilbik, Keller, and Alexander 2011), with some modifications to fit our context:

- y is the *parameter*.
- Q_i is the specific value that the *parameter* may take.
- P a summarizer, which is a tuple made up of the subject of description, *element* and a fuzzy predicate, e.g., low, high.

The predicates we use to pair with the element to make up the summarizer P, are the following: increasing, decreasing, highest, and lowest. The first protoform is for a trend in a time series:

$$Q_{min} \text{to } Q_{max} \text{has } P \tag{1}$$

where Q_{min} and Q_{max} are the beginning and end of the time series represented on the x-axis. The predicate values that are used in this protoform are increasing and decreasing.

The next protoform is a trend for an ordinal series for the x-axis:

$$(y, p) \text{ has } P \tag{2}$$

where y is the parameter that has an ordinal value and p is the trend of the ordinal value, which can be increasing or decreasing.

The next protoform is salience protoform:

$$y\ Q_i \text{ has rank } P \tag{3}$$

where the predicates of P are xth highest or lowest.

The last protoform is the max/min protoform:

$$y\ Q_i \text{ has } P \tag{4}$$

Here, the predicates are highest and lowest for the maximum and minimum values, respectively.

To generate the summary, the method follows the steps illustrated in Algorithm 1. As input, the method takes a semantic-graph representation of the chart and returns a text summary in the form of sentences stored in string S. Based on reading the labels associated with edges between the element and parameter nodes, the algorithm determines which protoform to use for generating the summary. One or more sentences can be generated and eventually combined to produce the final summary.

Evaluation

In order to assess the quality of the summaries, it is key to evaluate the main hypotheses the approach relies on to generate them. The summaries using this method follow a four step sequential pipeline, where each step requires certain input to be passed on to it from the previous step. Thus, we evaluate the final summary based on both the accuracy of the information it contains in addition to how well the summary captures the message the chart was designed to illustrate. To break this down, we describe in detail the results of: accuracy of the values produced by the chart data extraction module, accuracy of the results from the feature extraction module, and relevance of the selected features to the purpose of the chart.

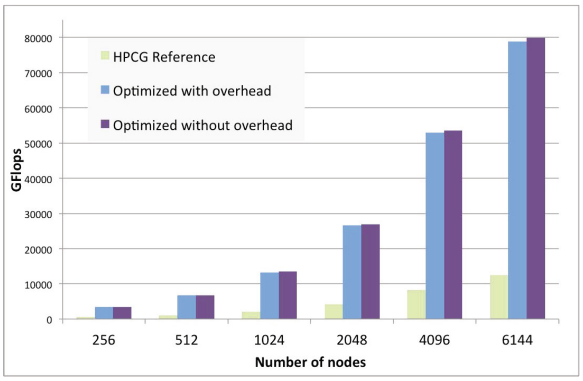

Figure 2: Sample Bar Chart from Random PDF Document.

Dataset

Our entire dataset consists of $40,000$ figures extracted from 10,000 articles published in top fifty computer science conferences between 2004 and 2014. The figures were extracted using a recently released system *pdffigures* (Clark and Divvala 2015). Their system produces an image file and a JSON metadata file for each figure in the document. The metadata contains the location of the bounding box of the figure within the document page and caption. For figures embedded as vector graphics, (eps/ps/PDF), the metadata also contains the text and the bounding box of the words inside the figure. We observed that around 50% of these figures contained sub-figures: often a combination of line graphs, bar charts, and pie charts. By manual inspection of the extracted images we found roughly 1,100 of these documents contained bar charts. By taking a random sample of these charts we found that 67% of them follow the assumptions followed in the data extraction step. To conduct our experiments we use a subset of 300 bar charts extracted from over 1,000 PDF files.

Chart-Data Extraction

Data extraction involves correct recognition of a chart's bars, y-axis scale, axes labels and names. We run experiments on the extracted charts and through manual inspection determine the accuracy of: bar recognition, y-scale calculation, extraction of x-labels and axes names, where they exist. Table 1 shows precision, recall, and type-ii error for these steps. As noted the recall is higher than precision for the bar extraction, which implies that the connected component method is effective in capturing the bar shaped components in the graphic. Bars that go undetected either have very small height, or are the same color as the background, e.g. white bars in white-background charts. However, lower precision implies that a simple heuristic based approach to classify these bar shapes as data bar or noise, leaves room for improvement.

Y-Scale Extraction The scale of the x-axis, which is the graphic-pixel to real-data ratio, can only be correctly calculated if both the y-axis labels are correctly located and if the OCR returns their correct text values. The locating of

	Precision	Recall	Type-II error
bar extraction	86%	89 %	11%
x-label region	83%	88%	12%
x-label text	73%	79%	21%
x-axis name characters	87%	86%	13%
x-axis name words	85%	82%	18%
y-axis name characters	77%	74%	26%
y-axis name words	79%	68%	32%

Table 1: Chart-Data Extraction Precision, Recall and Type-II Error

y-label regions -since it is computed as simply the region left to the leftmost bar- depends on the correct identification of the bars. Thus, we evaluate only the accuracy of the y-scale as either correct or incorrect, since it directly relies on a step that has already been evaluated. The method is able to identify 77% of y-scales correctly. This result is lower in charts extracted from PDF documents as opposed to charts from the web or ones from chart design tools (Al-Zaidy and Giles 2015). Many interesting factors contribute to this rate, such as the quality of the charts and accuracy of chosen OCR for the chart resolution.

X-Axis labels The x-axis labels are used to name the bars. Since many text strings can appear below the x-axis, we evaluate both the correct identification of the label's location and the correctness of the text. Table 1 shows the accuracies of extraction of the label region and the results for correct recognition of the text string it contains. Most cases where the method fails to identify the regions correctly is when the extracted bars contain many false positives. This will cause inaccuracies in the location of the x-axis, and consequently the x-axis labels. In it's current state the method assumes labels are horizontally aligned, thus if a single label spans more than one line, only the string in the first line will be returned. As for the text accuracy, the method does not address slanted labels, although their regions can be identified, the OCR does not recognize the texts. Additionally, if the labels are too small, the text resolution will not be high enough to be recognized.

Axes Names Table 1 shows the results for the extraction of the axes names. In many cases, the text characters are correctly extracted however the isotropic dilation may not be able to separate them into their individual words. Thus, we provide a detailed evaluation, one for the character recognition and another for word recognition. Intuitively, the word recognition, i.e. separating the characters into individual words, shows lower performance when the words are placed too close to one another. Sometimes when the image is scaled in one dimension more than the other before it is embedded in the PDF document, the character spacing becomes smaller than an average space for it's font size, which can affect the recognition results. However, the character recognition relies mainly on the OCR's performance. If a word has more than 90% of it's characters returned correctly it is considered correct. The character recognition was noted to perform with better accuracy when the font

used is not in bold, with a reasonable image resolution. Certain characters were commonly misinterpreted by the OCR, such as 'r' and 'Q' as well as special characters such as '%' and '#' signs.

Feature Extraction

The features extracted from the images are salience, trend, maximum and minimum, time-line and ordinality of the x-axis. Salience refers to when a specific bar has special design features that are meant to draw attention to it. Design features that can be used to express a bar's salience are giving it a different color than the other bars, or by annotating it with a text string. By examining the charts in our data set we found that less than 1% contain these salience features, which is not a sufficient size to evaluate the extraction of this feature.

To evaluate the identification of the trend feature, we first evaluate the extraction of the different bar series and groups. A bar series is the bars that belong to a single data series and are represented graphically by having the same color. A bar group is the collection of bars that have the same x-label. Evaluating these two extracted values is key because in our specific dataset of scientific charts the trends are mostly expressed over a data series or among a group. Since at this point we do not extract the legends of the chart, we determine the chart's trend based only on the trend of each data series, since the summary can be generated by using the x-axis name as the parameter. An appealing extension to this step is to extract the legends and their names so that they can be used as the parameter for the trend protoform, yielding an even more accurate description. Table 2 shows the results for the series and group identification steps. It is important to note that this step is based on the correct identification of the y-scale. Charts with a y-scale that was not extracted in the data extraction module, i.e. returned 'nil' by the extractor, cannot proceed to this step, thus the accuracy is calculated over only charts whose y-scale was extracted. Table 2 also shows the percentage of correctly identified maximum and minimum values. This is calculated over the entire set of bars in a chart. The fact that the bar extraction overlooks very small bars contributes to a great percentage of incorrect minimum values.

The last feature extracted is determining whether the data is a time series and whether the x-axis is a trend of ordinal values. Table 2 shows the percentage of correctly identified time series and ordinal x-axis values. Two factors affect the accuracy of this step: the first is the accuracy of the x-label extraction and correct recognition of their textual values. The second, is that the method in it's current design classifies the x-axis data as timeline only if all the x-axis labels are classified as such. However, the case where only a single label is incorrectly extracted is not uncommon and in the future this occurrence should be tolerated to achieve better detection rates.

Feature	Extraction Accuracy
bar series	100%
bar groups	96%
trend in series	81%
trend in group	79%
x-axis timeline	89%
x-axis ordinal	65%
maximum value	88%
minimum value	68%

Table 2: Feature Extraction Evaluation Results

	Summary Accuracy
Facts+	74%
Facts-	56%
Relevance	83%

Table 3: Generated Summary Evaluation

Summary generation

The final step is to evaluate the final summaries, which is an evaluation of the entire system performance. In order to determine the effectiveness of the summaries we evaluate the summaries based on two metrics: fact accuracy and summary relevance. To determine the fact accuracy we evaluate whether the information provided by the summary is simply correct or not. Facts+ indicates an optimistic evaluation, where minor errors that can be corrected by existing tools are overlooked, such as a single incorrectly retrieved character in a word. The Facts- measures the accuracy with a pessimistic evaluation where the summary facts are considered correct only if the data has been extracted with 100% accuracy. The relevance is based on whether the user thinks the summary captures the most important information the chart is designed to illustrate. However, this type of evaluation requires a quantification of the summary relevance in order to be able to measure it. In our evaluation, we assume a summary is considered relevant if it succeeds to mention either trend, maximum and minimum value over a timeline, or a trend in the ordinal values of the x-axis. Or if it successfully concludes the overall trend or max/min values whichever is present, in the case where the chart x-axis is neither a timeline nor contains ordinal value trend. For illustrative purposes, the summary that is generated by applying the method to the chart in Figure 2 is: *GFlops increases for increasing Number of nodes*.

Since the bar values exhibit a trend of increasing value and the x-axis values are ordinal values, the protoform 2 is applied here. It is noted though, that a large number of charts in scientific documents are used to illustrate experimental results. The results show that most summaries express trends and maximum and minimum values. This is consistent with the fact that not many contain salience features. Table 3 shows the result of user evaluation of the summaries generated by the charts. This step also was only evaluated for charts whose y-scale was extracted.

Conclusion

In this paper we propose a summary generation method for scientific charts, specifically bar charts. Our system uses image processing techniques to extract data from the charts and provides an analysis of the data. Charts are given semantic structure by means of semantic-graphs. The summary generation techniques are applied to the labeled graph to produce a textual description of the chart. A main challenge to this work is that the quality of the feature extraction is based on the accuracy of the data extraction method. Data extraction methods have indeed reached high accuracies on this front, however, if a single misread value propagates through the entire pipeline the summary may not be very representative of the chart's content.

The proposed framework is flexible to a myriad of extensions for improving the labeling and summaries. Identifying additional features can result in more complex, and perhaps, more accurate summaries. A notable finding from this study is that salience is not a common message in a data set such as the one used here, i.e. charts found in computer science scholarly documents. This indicates that certain types of messages can be more common in charts in certain contexts. A future extension for this work is to identify the messages that are more pertinent to charts found in computer science scholarly documents.

References

Al-Zaidy, R. A., and Giles, C. L. 2015. Automatic extraction of data from bar charts. In *Proceedings of the 8th International Conference on Knowledge Capture*, 30. ACM.

Chen, S. Z.; Cafarella, M. J.; and Adar, E. 2011. Searching for statistical diagrams. *Frontiers of Engineering, National Academy of Engineering* 69–78.

Chen, Z.; Cafarella, M.; and Adar, E. 2015. Diagramflyer: A search engine for data-driven diagrams. In *Proceedings of the 24th International Conference on World Wide Web Companion*, 183–186. International World Wide Web Conferences Steering Committee.

Chester, D., and Elzer, S. 2005. Getting computers to see information graphics so users do not have to. In *Foundations of Intelligent Systems*. Springer. 660–668.

Clark, C., and Divvala, S. 2015. Looking beyond text: Extracting figures, tables, and captions from computer science paper. In *AAAI Workshop on Scholarly Big Data*.

Demir, S.; Oliver, D.; Schwartz, E.; Elzer, S.; Carberry, S.; and McCoy, K. F. 2010. Interactive sight into information graphics. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, 16. ACM.

Demir, S.; Carberry, S.; and McCoy, K. F. 2008. Generating textual summaries of bar charts. In *Proceedings of the Fifth International Natural Language Generation Conference*, 7–15. Association for Computational Linguistics.

Elzer, S.; Carberry, S.; and Zukerman, I. 2011. The automated understanding of simple bar charts. *Artificial Intelligence* 175(2):526–555.

Fang, J.; Mitra, P.; Tang, Z.; and Giles, C. L. 2012. Table header detection and classification. In *AAAI*.

Ferres, L.; Verkhogliad, P.; Lindgaard, G.; Boucher, L.; Chretien, A.; and Lachance, M. 2007. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, 67–74. ACM.

Huang, W., and Tan, C. L. 2007. A system for understanding imaged infographics and its applications. In *Proceedings of the 2007 ACM symposium on Document engineering*, 9–18. ACM.

Kataria, S.; Browuer, W.; Mitra, P.; and Giles, C. L. 2008. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *AAAI*, volume 8, 1169–1174.

Li, Z.; Carberry, S.; Fang, H.; McCoy, K. F.; Peterson, K.; and Stagitis, M. 2015. A novel methodology for retrieving infographics utilizing structure and message content. *Data & Knowledge Engineering* 100:191–210.

Liu, Y.; Bai, K.; Mitra, P.; and Giles, C. L. 2007. Tablerank: A ranking algorithm for table search and retrieval. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, 317. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Lu, X.; Kataria, S.; Brouwer, W. J.; Wang, J. Z.; Mitra, P.; and Giles, C. L. 2009. Automated analysis of images in documents for intelligent document search. *International Journal on Document Analysis and Recognition (IJDAR)* 12(2):65–81.

Savva, M.; Kong, N.; Chhajta, A.; Fei-Fei, L.; Agrawala, M.; and Heer, J. 2011. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 393–402. ACM.

Tuarob, S.; Bhatia, S.; Mitra, P.; and Giles, C. L. 2013. Automatic detection of pseudocodes in scholarly documents using machine learning. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 738–742. IEEE.

Wilbik, A.; Keller, J. M.; and Alexander, G. L. 2011. Linguistic summarization of sensor data for eldercare. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2595–2599. IEEE.

Yang, L.; Huang, W.; and Tan, C. L. 2006. Semi-automatic ground truth generation for chart image recognition. In *Document Analysis Systems VII*. Springer. 324–335.

Zadeh, L. A. 2002. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. In *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*, 523–525. IEEE.