# RECOGNIZING THE INTENDED MESSAGE OF LINE GRAPHS: METHODOLOGY AND APPLICATIONS

by

Peng Wu

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Winter 2012

# RECOGNIZING THE INTENDED MESSAGE OF LINE GRAPHS: METHODOLOGY AND APPLICATIONS

by

Peng Wu

Approved: _____

Errol L. Lloyd, Ph.D.
Chair of the Department of Computer and Information Science

Approved: _____

Babatunde A. Ogunnaike, Ph.D.
Interim Dean of the College of Engineering

Approved: _____

Charles G. Riordan, Ph.D.
Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy in Computer
Science.

Signed: _____

Sandra Carberry, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy in Computer
Science.

Signed: _____

Stephanie Elzer, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy in Computer
Science.

Signed: _____

Daniel Chester, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy in Computer
Science.

Signed: _____

Vijay K. Shanker, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy in Computer
Science.

Signed: _____

Ben Carterette, Ph.D.
Member of dissertation committee

# ACKNOLWEDGEMENTS

I want to thank my parents and my grandmother, for always being there. Without their love, support, tolerance and advice, I would never have gone so far.

I would like to express my deepest thanks to my advisor, Dr. Sandra Carberry, for the great guidance she provided during my Ph.D. years. Her enthusiasm, intelligence and diligence in research is always a model for me to follow in my future career.

I would also express my special thanks to all my colleagues in HLT-NLP lab, including Dr. Kathy McCoy, Rich Burns, Charlie Greenbacker, Praveen Chandar, and Keith Trnka. They have built a lovely and warm environment for my research, from green house to tea house. Our many lunches were the most fun time during those days. I would also sincerely thank Dr. Stephanie Elzer and her students from Millersville University for contributing a lot of invaluable ideas and data to bring my research from concepts into experiments.

In the end, I want to thank National Science Foundation and National Institute on Disability and Rehabilitation Research. My research was supported by National Science Foundation funding under Grant No. IIS-0534948 and III-1016916, and by the National Institute on Disability and Rehabilitation Research under Grant No. H133G080047.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Information graphics (line graphs, bar charts, etc.) are common in popular media and periodicals. They are usually included in such documents to convey a message. This dissertation discusses the processing of one kind of information graphic, namely a line graph. It presents a learned model for segmenting a line graph into visually distinguishable trends and a Bayesian network inference model that hypothesizes the intended message of the graph based on communicative signals in the graphic. Besides recognizing the intended message of line graphs, this dissertation also presents a method for identifying the paragraph in the document that is most relevant to its information graphic. The research results provided by this dissertation can be used for several purposes: to give blind individuals access to information graphics in an article, to provide the basis for a longer summary of the graphic, to build a summary that captures both the article and its containing information graphics, and to indicate a graphic's content when indexing it for retrieval in a digital library.

# Chapter 1

# INTRODUCTION

Information graphics are non-pictorial graphics such as bar charts and line graphs. They appear in popular media such as *New York Times*, *Businessweek*, *Wall Street Journal*, etc. as well as in scientific articles. Usually they are one part of a multimodal document which contains both the textual article and the information graphic. Information graphics in popular media differ from ones in scientific articles in several ways. First, graphics in scientific articles are intended to display data and facilitate an analysis of it. Second, the text of a scientific article generally refers explicitly to graphics with referents such as "see Figure X" and often contains an explanation of the data depicted in its graphs. On the other hand, the text of a document in popular media rarely references its graphics explicitly and often says nothing about the content of its graphics.

The overwhelming majority of information graphics in popular media, such as newspapers and magazines, have a message that they are intended to convey. For example, the line graph in Figure 1.1 appeared in *USA Today* and ostensibly is intended to convey the message that there has been a recent decrease in box office gross revenue in contrast with the preceding rising trend. And the line graph in Figure 1.2 ostensibly is intended to convey a changing trend in ocean levels from relatively stable between 1900 and 1930 to rising thereafter. We contend that a graphic's intended message constitutes a brief summary of the graphic's high-level content and captures how the graphic should be "understood".

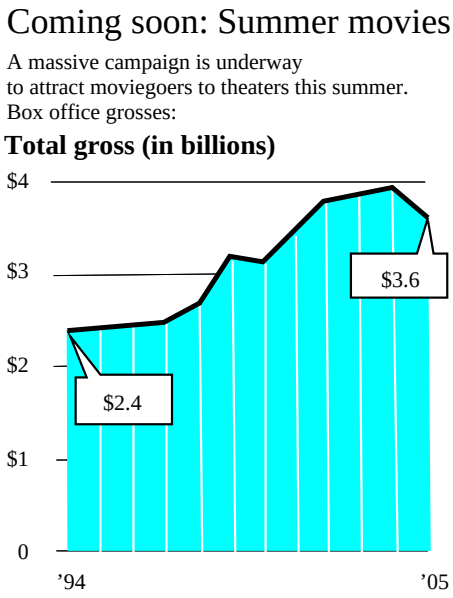



Figure 1.1: A line graph from *USA Today* with the message that there has been a recent decrease in box office gross revenue in contrast with the preceding rising trend

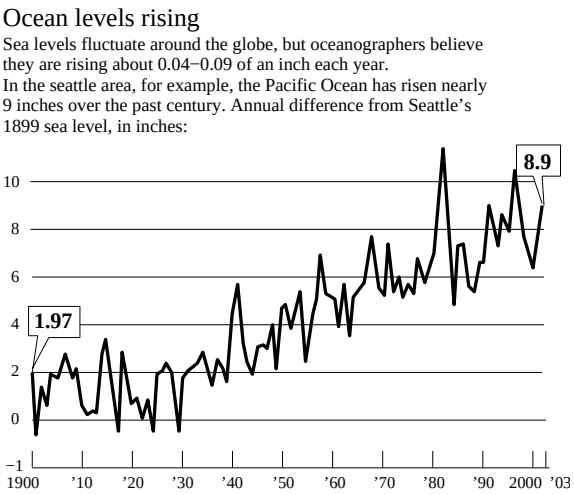



Figure 1.2: A line graph from *USA Today* conveying a changing trend in ocean levels from relatively stable between 1900 and 1930 to rising thereafter

Our goal is the development of a system that can recognize the intended message of a line graph. The message recognition system could play an integral role in three very different projects:

1. Many assistive technology projects designed for blind individuals are able to read the text on the screen to users. But they have difficulty conveying the graphics in multimodal documents. Although some images like a photograph of the subject might be safely ignored, information graphics cannot be ignored since they often support or complement the content of the text in the document. If the intended message of a graphic can be identified, then it can be used as the core of a summary of the graphic that is constructed and read to the user by an assistive technology system. Chapter 9 discusses how we have incorporated our system for recognizing the intended message of a line graph into SIGHT[31, 32], a system for providing

blind individuals with access to information graphics.

2. Currently most document summarizers consider only the text of an article when producing a summary even for a multimodal document. To incorporate the high-level content of the information graphics in the summary and generate a rich summary of the multimodal document, a message recognition system could first identify the intended message of the graphic, and an English rendition of this message (or a longer summary of the graphic based on this message) could be inserted into the article at the most appropriate place. Then extractive summarization techniques could be used to treat the flattened document as a whole and generate a summary from it.

3. Traditional information retrieval systems have focused on text retrieval and, to a lesser extent, image retrieval. But they are ineffective at retrieving information graphics. For example, when we search "graph CBS revenue against other television networks" in Google image search, none of the 20 results on the first page is relevant. Most of the top results are tables instead of information graphics. Although some results are indeed bar charts, they are either not about CBS or not about revenue. This is because the image search for this type of query relies heavily on the keywords matching words in the article instead of trying to recognize what is conveyed in the image.

   We thus hypothesize that a graphic's intended message should play a major role in deciding whether to retrieve the graphic in response to a user query. The message recognition system can identify the high level content of the information graphic first and then the query can be compared with the intended message to decide whether to retrieve it. For example, a query such as "When did Nokia sales start to fall?" requires the system to recognize that the query is seeking graphs with a Change-Trend message, identify such graphics for Nokia sales, and then provide the

results matching the query. Thus we believe that a successful retrieval system will need to index information graphics by not only the words in its caption/description but also by its intended message.

This dissertation presents a new methodology for inferring the intended message of a line graph in a multimodal document. In previous research[11, 30], Elzer developed a system for identifying the intended message of a simple bar chart. However, line graphs differ from bar charts in several ways that significantly impact the required processing. First, line graphs are the preferred medium for conveying trends in quantitative data over an ordinal independent axis[57]. Second, as our corpus studies demonstrate, the kinds of messages conveyed by line graphs differ from those conveyed by simple bar charts. For example, the line graph in Figure 1.3 ostensibly is intended to convey a sudden big drop in Afghanistan's opium crop that is not sustained; in our research, we have not encountered a bar chart that conveys a message of this type. Third, although line graphs and bar charts share some of the same kinds of communicative signals, line graphs use other communicative signals that are not found in bar charts, such as the length of the ending segment. Fourth, recognition of the message conveyed by a line graph must capture the viewer's tendency to perceive it as a sequence of visually distinguishable trends rather than as a set of discrete data points. Thus we need a method for identifying these trend segments. Moreover, these latter two factors necessitate a different structure and different processing for the message recognition system than was used for bar charts where recognition relied heavily on perceptual task effort.

This dissertation also provides a method for identifying the paragraph in a multimodal document that is most relevant to its constituent information graphic. We hypothesize that this is important for assistive technology that conveys graphics to blind users, for summarization of multimodal documents, and for the retrieval of information graphics:

1. In providing blind users with full access to multimodal documents, we hypothesize

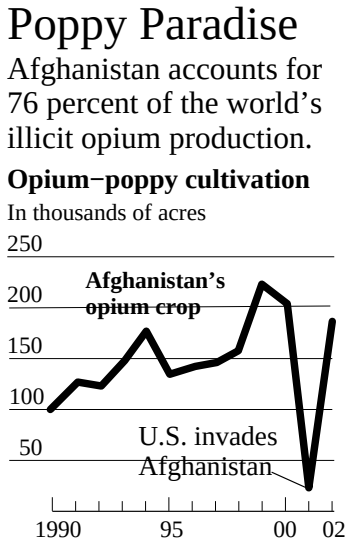



Figure 1.3: A line graph from Newsweek which conveys a sudden big drop in Afghanistan's opium crop that is not sustained

that the brief summary of an information graphic should be read at the most coherent place in the document's text – that is at the paragraph that is most relevant to the graphic. Otherwise, the purpose of assisting blind users to achieve better understanding of the multimodal document may be compromised. Our analysis has shown that line graphs generally don't occur adjacent to the most relevant paragraph, so that reading the content of the information graphics at the point where the graphic is located won't necessarily result in a coherent presentation.

2. Extractive summarization techniques produce a summary by extracting multiple sentences from an article and organizing them together as the summary of the article. Some heuristics are used to decide which sentences to extract, such as measuring whether a sentence is the first sentence of a paragraph and whether two sentences are coming from the same paragraph, etc. These heuristics assume that the article is coherently organized. Thus to generate a coherent summary of a multimodal document using extractive techniques requires that a coherent document be passed as input to the summarization system; this requires that we first insert

the brief summary of the information graphic into the paragraph where it is most relevant.

3. To develop a system for retrieving information graphics from a digital library, we hypothesize that it will be necessary to take into account both the information inside a line graph and the information inside the article containing the line graph. However, the article containing a graphic may cover multiple topics and not all of them will be relevant to the graphic. Therefore if the retrieval system considers only paragraphs relevant to a graphic in deciding whether to retrieve the graphic, it will avoid extraneous text and presumably produce better results.

This dissertation is organized as follows. Chapter 2 presents related work on graph design, comprehension, generation and summarization. Chapter 3 through Chapter 7 present the details of the intended message recognition system for line graphs. Chapter 8 discusses our method for identifying the paragraph in a multimodal document that is most relevant to a line graph. Chapter 9 discusses our contributions to SIGHT, an assistive technology project which incorporates our research and provides a blind user with a brief summary of a line graph at the most relevant paragraph in a multimodal document. Chapter 10 presents ideas for future work. This dissertation ends with Chapter 11 which summarizes our research and its contributions.

# Chapter 2

# RELATED WORK

Various communities have pursued research on the analysis, design, and understanding of graphs. There are resources discussing how to generate a graph and how different elements in a graphic can help the graph designers achieve their communicative goals. Researchers from cognitive science have analyzed how people comprehend a graph such as a bar chart or a line graph and what is the procedure that readers go through while viewing a graphic. Some research efforts have focused on generating graphs that achieve a communicative goal. Other groups of researchers have attempted to develop machines that understand or at least extract information from an information graphic. Researchers from the natural language processing area have also tried to develop software for summarizing a graph in natural language.

This chapter is organized as follows. Section 2.1 covers related work on designing a graph. Section 2.2 discusses related work from cognitive and psychological research on comprehension of a graph. Section 2.3 presents work on information graphics generation. Section 2.4 discusses existing work on understanding an information graphic and recognizing its intended message, and also discusses existing work on summarizing graphs in natural language.

## 2.1 Graph design

To analyze how people or even a computer can comprehend and understand an information graphic, it is useful to first view the graph from a designer's perspective. As

discussed in the book "Elements of Graph Design"[49], many graphs are designed without consideration for principles of human visual perception and cognition, and thus limit the effective communication between designers and readers. A graphic designer should not throw arbitrary information into a graphic, but instead should follow some rules to make it easy for the readers to comprehend the pattern, trend or comparison. These design criteria suggest important aspects of a graph understanding system.

1. "human minds are not cameras"[49]. Humans tend to interpret visual elements by grouping them by proximity or similarity. According to this principle, a graph designer should put similar elements together to reduce the reader's cost of comprehension. In our research, we reduce a large number of data points sampled in a line graph to a small group of visually distinguishable trends.

2. The human visual system and memory system tend to make a direct connection between the properties of a pattern and the properties of the entities symbolized by that pattern. For example, humans connect a picture of an elephant with the word "elephant". According to this principle, a designer might use "fall" or "drop" in a graphic's caption as a symbol to emphasize a falling trend conveyed in a graph. We take this into account by incorporating such clue words into our recognition system. Kosslyn[49] also explicitly discusses the caption which he states is a comment on the display, a short description that explains key terms, or text that directs the reader's attention to specific features of the display. Thus it follows that the graphic designer may use the caption to facilitate the understanding of a line graph, so that a caption might not only indicate the subject of a line graph but also contain clues for the reader to better comprehend the graph.[1] In our research, the caption contributes two features that are used to understand the graph: the verb/adjective category in the text of the caption and the entity salience brought about by words

[1] Unfortunately, many captions in popular media are very short and unhelpful.

in the caption – that is, a noun in the caption which refers to a point in the graphic, thereby making that point salient.

3. Humans can keep only a certain amount of information in mind at any one time. A graph should not require the reader to hold more than four perceptual groups in mind at once. According to this principle, we hypothesize that graph designers are unlikely to have communicative goals that would force the human viewer to memorize more than four distinct segments; thus none of our 10 categories of possible messages, such as Change-Trend, contains more than four segments.

In providing guidance for designing line graphs, several common practices are also recommended, such as labelling critical points explicitly with their values when a few specific point values are to be emphasized and avoiding the labelling of specific points unless they are particularly important. This recommendation supports our hypothesis that such annotations are intended by the graph designer to facilitate recognition of the graph's intended message and thus are communicative signals that should be utilized in our graph understanding system. In the examples in the following chapter, we will show how the location and number of annotations change how the system recognizes the intended message of a line graph.

These discussions about the design of information graphics give us insights into how graph designers use different communicative signals in presenting the data and achieving their communicative intention. These are exploited in our graph understanding system.

## 2.2 Graph comprehension

The graph design research discussed in the preceding section presents principles and recommendations on graphics from a designing perspective. Many other research efforts[81, 71, 45, 82] have investigated graphs from a comprehension perspective.

Both Shah[82] and Pinker[71]'s work investigated the procedure for comprehending a graph. First, a viewer must capture the visual descriptions, which are the visual features inside the graphic (such as a curved line). Second, a viewer must relate the visual features to the conceptual relations that are represented by those features. In Pinker's work, this process is described as a graph schema. A graph schema organizes the visual descriptions and maps the visual descriptions to an interpretation. In our research, we assume that some types of visual descriptions(visual features) for line graphs are identified and recorded in an XML schema. The XML representation contains all the necessary primitive visual descriptions such as caption, label, data point coordinates, etc. The statement in Shah et al.'s work[12, 82, 102] that viewers are more likely to describe x-y trends when viewing line graphs than when viewing bar graphs also supports our method of segmenting a line graph into a series of visually distinguishable trends.

Besides graph comprehension, Pinker's work also discussed four procedures required for readers to make predictions. They are: a *MATCH* process that recognizes individual graphs as belonging to a particular type, a *message assembly* process that creates a conceptual message out of the instantiated graph schema, an *interrogation* process that retrieves or encodes new information on the basis of conceptual questions, and a set of *inferential processes* that apply mathematical and logical inference rules to the entries of the conceptual message. Although in our current research we don't do prediction on line graphs, it would be possible to extend our work to Pinker's procedure. Such predictions might be useful for blind users and for question answering systems. This potential extension will be discussed in Chapter 10 as future work.

Schnotz's book[81] is a collection on many topics about graph design and graph comprehension. It includes research work on graphical codes, graphics processing and graphics representations. The chapter by Maichle[64] talked about the cognitive processes in understanding line graphs. He discussed individual differences when different readers try to comprehend the same line graph , as a result of readers having different

graph schemas in mind. A reader with sophisticated domain knowledge has a better line graph schema than other readers who use only a general line graph schema. Regarding line graphs used in popular media such as newspapers and popular magazines, we hypothesize that the graphic designer doesn't require readers to use specific graph schemas to comprehend them. This hypothesis supports our approach of training a Bayesian network with a set of line graphs to learn a model and testing it on another set of line graphs, which can be regarded as learning a general graph schema from a set of common line graphs. Otherwise, we would need to design individual graph schemas for each different domain and only process a line graph within its already-known domain.

## 2.3 Graph and caption generation

Mittal et al.[67, 66] presented three strategies for generating explanatory captions to accompany information graphics and implemented them in a caption generation system. In their system, the graphical displays are designed by an automatic presentation component *SAGE* and are often complex because they typically display many data attributes at once. In their research, the information graphics are mostly composite graphs, which is a combination of multiple attributes together in the same display. It is denoted as "one space" if the graphic shows one relationship and denoted as "multiple spaces" if the graphic shows multiple dependent attributes. For example, the two spaces graph may have the same independent variable as the y axis and two dependent variables displayed side by side. Mittal's caption generation system was illustrated by generating explanatory captions for a range of graphics from a data set about real estate transactions in Pittsburgh. It employed three main strategies. The first strategy applied when the graphic had only one space and the independent attribute was along one of the axes, or when there were multiple spaces and the independent attribute was mapped to the axis of alignment. In this case, the explanation should reinforce the organizing role of the functionally independent attribute. The second strategy applied when there was only one space in the graphic; in that case, the explanation emphasized the relation between the attributes encoded against

the axes. The third strategy applied if the graphic had multiple spaces and the axis of alignment encoded a dependent attribute. In this case, the explanation described each space independently, using the appropriate strategy for each space.

Mittal et al.[67, 66] have adapted and integrated work in natural language generation in different subareas: text planning, aggregation, centering, computing referring expressions, example generation, and linearization. In addition to these NLG techniques, their generation of the textual captions for information graphics required knowledge sources such as a representation of the syntax of graphical displays which is the structural, spacial and other relations among graphical objects and their properties; a representation of the semantics of graphical displays; and a mechanism for determining which aspects of graphical displays must be explained based on their perceptual complexity.

Our work focuses on recognizing the intended message of line graphs which can be used as the core of a natural language summary. Although Mittal et al.'s work has a Complexity Metric Module which measures multiple complexities, it doesn't capture the intended message of the information graphics. It is partially a result of the information graphics they processed being generated from *SAGE* which uses a variety of graphical techniques to integrate multiple data attributes in a single display. But the emphasis of Mittal's work was the generation of captions that explained the attributes depicted in the graph and their notes.

The AutoBrief project[38] designed a system which can generate a multimodal document consisting of text and information graphics. AutoBrief includes a Presentation Generator, which plans presentations that consist of communicative goals and designs text and information graphics to achieve these communicative goals. In our research, we are not generating information graphics; instead we try to capture the communicative goal conveyed by the graph designer by analysing the graph's communicative signals, which is the inverse of the graph generation process.

## 2.4 Graph understanding and summarization

In addition to investigating the design and comprehension of line graphs from a cognitive or psychological perspective, researchers in artificial intelligence and natural language processing have pursued research aimed at giving machines the ability to understand and summarize a graph.

Ehud Reiter's team has undertaken multiple projects on generating summaries from non-linguistic input data[77, 83]. This is relevant to our work since they consider patterns in the data and understanding line graphs also requires identifying patterns. Their project on constructing textual summaries of time-series data sets for gas turbine engines[99, 100] detects big spikes in gas turbine sensor readings and generates textual summaries of the spikes. Their project for weather forecasts[78] matches weather features such as wind strength, direction, and visibility with different textual patterns and generates a textual summary. Their BabyTalk project[73] applied their method to neonatal intensive care data to generate a textual summary of 45 minutes of continuous physiological signals and discrete events. The essential ideas behind their multiple projects is to capture the most salient patterns inside the data of a specific domain, design the textual summarization framework based on domain knowledge, and combine the captured pattern within a summarization framework to generate the final textual summaries. Our project and Ehud Reiter et al.'s project differ in three respects:

1. They work with machine-generated data, not a human-generated graph, and thus don't have an intended message. They only consider the salient data patterns such as a spike or a valley and do not consider the data series as a whole. Our project recognizes the intended message conveyed by a graph, based on the kind of graph selected, the data patterns, and the communicative signals in the graph.

2. Their project only deals with either time series data such as gas turbine sensor readings and neonatal intensive care data signals, or categorical data such as weather features. They rely on only the numeric data but our project has to consider various

communicative signals and combine all of the communicative signals together to hypothesize the graphic's message.

3. Their research relies on domain knowledge, so for different domains, different processing and methods are required. Our research must consider data from any domain where the graph designer has a message that he/she intends to convey, and thus we cannot rely on domain knowledge.

Ferres et al.[33, 50, 34, 35] constructed a system named iGraph which can provide short verbal descriptions of the information depicted in graphs; the system includes a way of interacting with it to request specific information. Their project shares many similarities with our project. First, their system also detects trends, but the trends are only limited to overall upward trend or downward trend, which is achieved by simply comparing the two end-points of the time series data and doesn't attempt to detect intermediate trends as we do. Second, their project can generate a textual description of a line graph but the description basically describes the characteristics of the line graph such as the label on the x axis, the maximum/minimum value on the y axis, and the pointwise trend between each pair of adjacent points. So although their project is also trying to build a textual summary of a line graph, it works at a low level and focuses on the raw data that's depicted in the graph. On the contrary, our project recognizes the intended message of a line graph and thus captures the high-level content of the graphic.

Elzer et al.[11, 30, 14] was the first to investigate recognition of the intended message of a graphic. Her work was limited to bar charts. She put the possible messages into categories and used a Bayesian network as the main framework. She also uses evidence nodes to represent the communicative signals for or against an intended message. Line graphs differ significantly from bar charts, as do the processes required for understanding them.

1. A line graph consists of a very large number of sampled data points. Even if these are combined to form a sequence of short jagged line segments, the set of line

segments will still be very large and at too low a level to represent the kind of visual features posited by Shah[82] and Pinker[71] for graph comprehension. Thus a line graph must first be split into a sequence of visually distinguishable trends, thereby converting the line graph into a set of discrete entities perceived by humans.

2. The intended message categories for line graphs differ from these for bar charts. Bar charts can have a "Rank" message which conveys the rank of a particular object in the whole set of objects but line graphs don't have this kind of intended message. On the other hand, line graphs have intended messages such as Contrast-Trend-Last-Segment which contrasts a potential new trend at the end of a line graph with the long trend preceding it, and Big-Jump which conveys an outstanding sharp rising spike in the data. These intended message categories are not conveyed by bar charts. We will discuss the intended message categories for line graphs in Section 3.

3. The communicative signals used in line graphs also differ from those in bar charts. Bar charts use the relative effort of perceptual tasks that the viewer might perform on the graphic as communicative signals[29], whereas line graphs use the features extracted from the visually distinguishable trends as communicative signals. We will discuss the communicative signals for line graphs in Section 4.

However, Elzer's work[11, 30, 14] has heavily influenced our work on line graphs. We also use a Bayesian network as our inference mechanism and we adopt her overall approach of extracting communicative signals from a graphic and using them as evidence about the graphic's intended message.

### 2.5 Summary

Researchers have investigated the design of information graphics and the process of human comprehension. Other researchers have explored the automatic generation of information graphics that achieve a specified communicative goal. Elzer was the first to

devise a system for understanding a bar chart by recognizing the message it was intended to convey. Although line graphs differ significantly from bar charts, our work on recognizing the intended message of a line graph draws on the methodology espoused by Elzer and exploits principles of graph comprehension identified by cognitive psychologists.

# Chapter 3

# KINDS OF MESSAGES CONVEYED BY LINE GRAPHS

We collected a set of simple line graphs from various popular media, including magazines such as *Newsweek*, *Time*, and *Businessweek* as well as local and national newspapers. We limited our collection to simple line graphs consisting of a single set of connected line segments. In addition, we have excluded line graphs that have y-axis tick marks on a non-linear scale (such as a set of uniformly spaced tick-marks labelled as $10$, $10^2$, $10^3$, etc.) because the interpretation of these line graphs may require domain knowledge such as a logarithm transformation as often required in scientific articles. We were interested in line graphs that ostensibly have a high-level message as opposed to graphics that just present data. Thus we did not include graphics such as ones depicting the hourly change in the Dow Jones Industrial Average which appear daily in the business section of daily newspapers or some of the *Businessweek* graphs which have a standard presentation and only display price data on a stock without any attempt to convey a message.

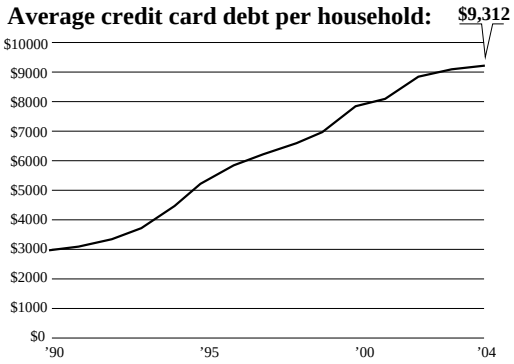
From this set of line graphs, we identified a set of 10 high-level message categories that we believe capture the kinds of messages that are conveyed by a simple line graph. The next section presents these message categories along with illustrative examples.

## 3.1 Message categories

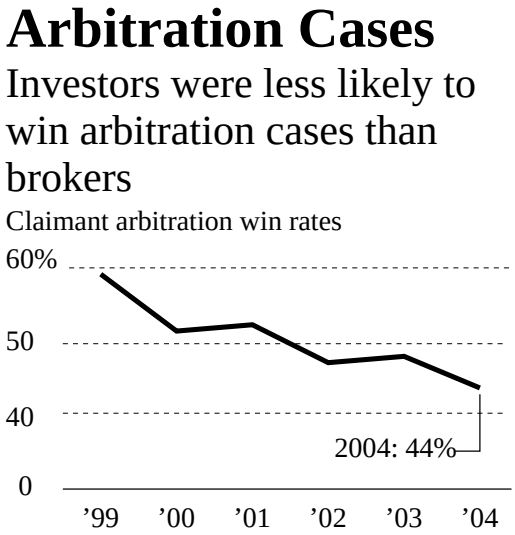
We analyzed the initial set of 350 line graphs to identify a set of message categories for line graphs. Each message category has a set of parameters that must be instantiated to represent an actual message in that category. Table 3.1 presents these message categories

| Intention Category | Description |
|---|---|
| **RT: Rising Trend** | There is a rising trend from <param$_1$ > to <param$_2$ > |
| **FT: Falling Trend** | There is a falling trend from <param$_1$ > to <param$_2$ > |
| **ST: Stable Trend** | There is a stable trend from <param$_1$ > to <param$_2$ > |
| **CT: Change Trend** | There is a <slope$_2$ > trend from <param$_2$ > to <param$_3$ > that is significantly different from the <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CTLS: Change Trend Last Segment** | There is a <slope$_2$ > segment from <param$_2$ > to <param$_3$ > that is not long enough to be viewed as a trend but which is different from the <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CTR: Change Trend Return** | There is a <slope$_3$ > trend from <param$_3$ > to <param$_4$ > that is different from the <slope$_2$ > trend between <param$_2$ > and <param$_3$ > and reflects a return to the kind of <slope$_1$ > trend from <param$_1$ > to <param$_2$ > |
| **CSCT: Contrast Segment Change Trend** | There is a <slope$_3$ > segment from <param$_3$ > to <param$_4$ > that is not long enough to be viewed as a trend but which suggests a possible return to the kind of <slope$_1$ > trend from <param$_1$ > to <param$_2$ > which was different from the <slope$_2$ > trend from <param$_2$ > to <param$_3$ > |
| **BJ: Big Jump** | There was a very significant sudden jump in value between <param$_1$ > and <param$_2$ > which may or may not be sustained |
| **BF: Big Fall** | There was a very significant sudden fall in value between <param$_1$ > and <param$_2$ > which may or may not be sustained |
| **PC: Point Correlation** | There is a correlation between the annotated points $< p_1 >, ..., < p_n >$ and the events referenced by the annotations $< a_1 >, ..., < a_n >$ |

Table 3.1: Categories of High Level Messages for Line Graphs




(a) A Rising-Trend line graph showing Debt rose from 1990 to 2004



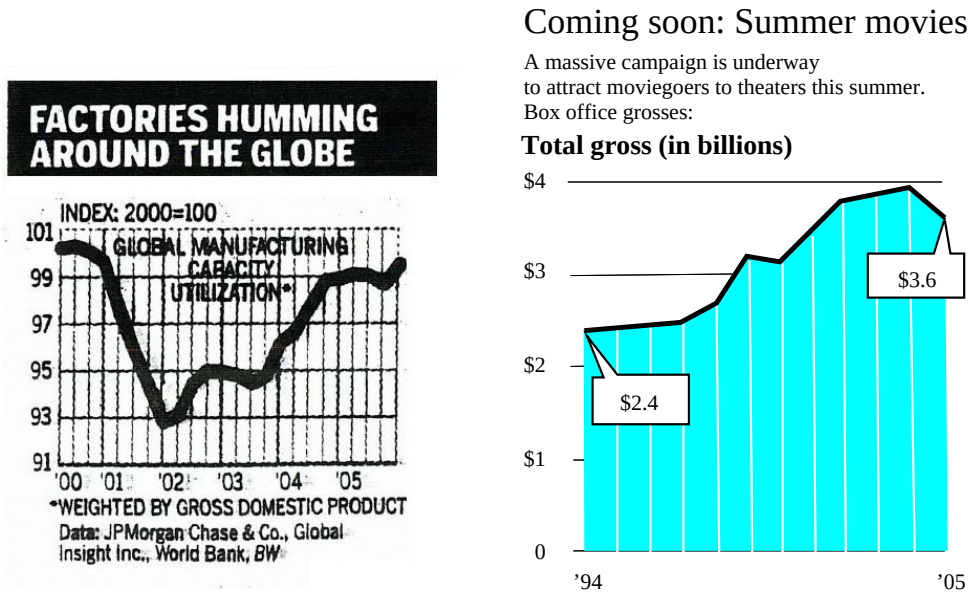

(b) A Falling-Trend line graph showing the claimant arbitration win rates fell from 1999 to 2004

Figure 3.1: Examples with Rising-Trend and Falling-Trend intended messages

along with English glosses describing what is conveyed by an instantiated message in that category.

The first three (Rising-Trend, Falling-Trend and Stable-Trend) capture line graphs that are intended to convey a single overall trend. They differ only in whether the trend represents an increase, decrease, or lack of change over the ordinal independent axis. These three categories of messages each take two parameters: the starting point of the trend and the ending point of the trend. Figure 3.1 shows line graphs conveying Rising-Trend and Falling-Trend intended messages.

Some line graphs are intended to convey a contrast between two trends, which change directions at some point. It might be a continuous recovery from a bad situation(a falling segment followed by a rising segment) or a long term under-perform after a long success before it(a rising segment followed by a falling segment). We refer to this message category as Change-Trend. Figure 3.2a shows a line graph with a Change-Trend message, namely that global manufacturing capacity utilization dropped from 2000 to 2002 and




(a) (b)

Figure 3.2: Line graphs with Change-Trend intended message on the left and with Change-Trend-Last-Segment intended message on the right

then rose from 2002 to 2006.

Sometimes the change from a trend might consist of a short segment at the end of the graph. Our analysis of line graphs suggests that this short segment is intended to convey the possibility of a changing trend but one that can only be confirmed by more data. For example, after a long term employment rate decrease, the employment rate might have increased for two months, but it is unclear whether this represents a new trend or is just a short aberration. Such situations often occur when the end of the graph captures data at the time that the graph was published. For example, a company with long term success might have a surprising decrease in revenue when the article is published, and the graph designer is warning investors that the depression **may** continue. Thus we have chosen to capture this in a separate message category which we call Change-Trend-Last-Segment. Figure 3.2b shows a line graph with a Change-Trend-Last-Segment, which shows that movie theater gross revenue rose from 1994 to 2004 and drops to 2005. Both Change-Trend and Change-Trend-Last-Segment take three parameters: the starting point,

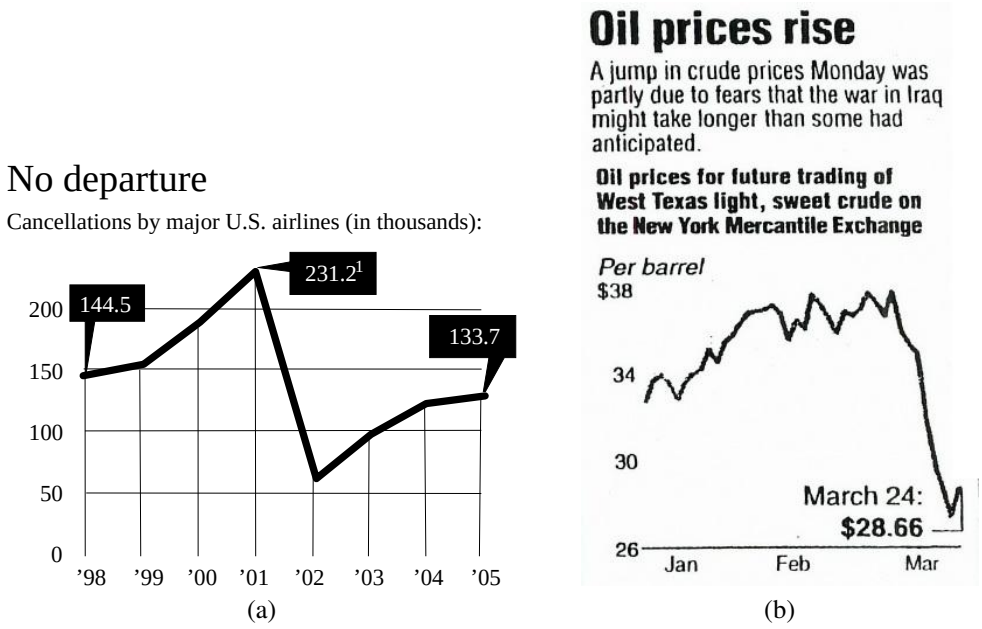



Figure 3.3: Line graphs with Change-Trend-Return intended message on the left and with Contrast-Segment-Change-Trend intended message on the right

the changing point and the ending point.

Some line graphs convey a message involving three trends. We refer to this message category as Change-Trend-Return. For example, Figure 3.3a shows that the cancellations by major U.S. airlines rose from 1998 to 2001 and then dropped to 2002, but rose again until 2005. However, as was the case with the message category Change-Trend-Last-Segment, the initial and contrasting trends may be followed by a short segment that only suggests the possibility of a return to the initial trend. Thus we included another message category which we refer to as Contrast-Segment-Change-Trend. Figure 3.3b presents a graph whose message is ostensibly that oil prices may be rising again following a stumble, which interrupted a previous rising trend. Both Change-Trend-Return and Contrast-Segment-Change-Trend take four parameters: the starting point, the first changing point, the second changing point and the ending point.

Although the differentiation of Change-Trend-Last-Segment from Change-Trend and Contrast-Segment-Change-Trend from Change-Trend-Return makes the recognition

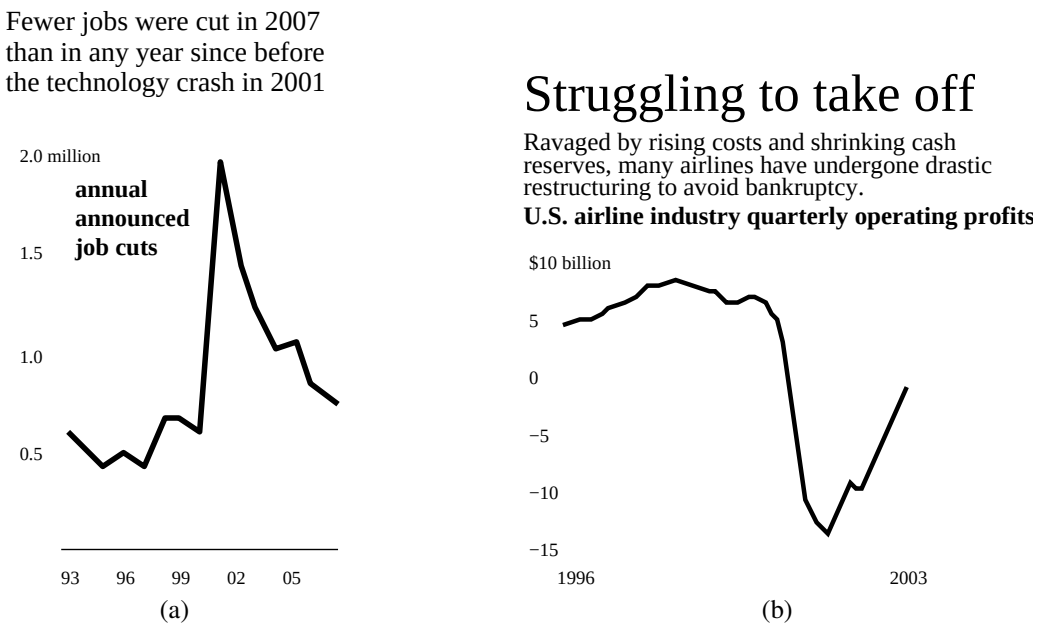
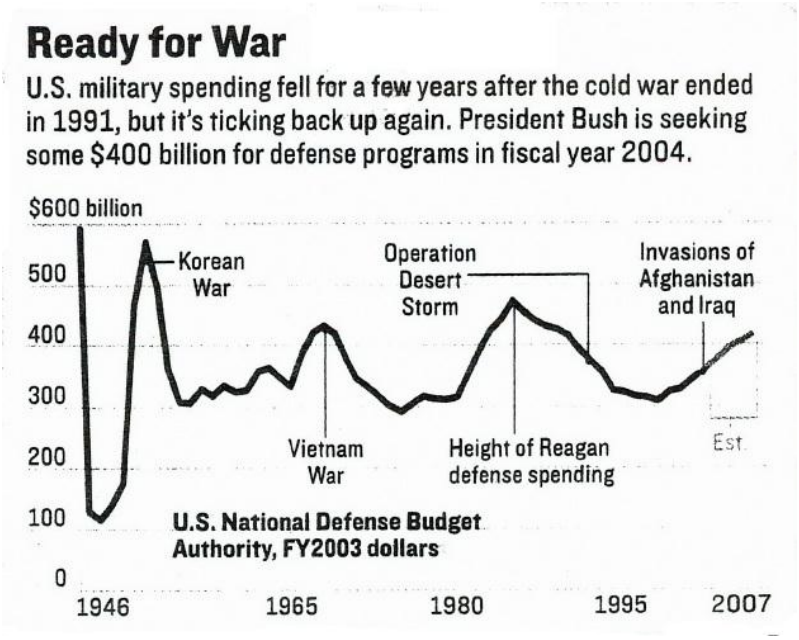



Figure 3.4: Line graphs with Big-Jump intended message on the left and with Big-Fall intended message on the right

of intended message more complex, we believe it better captures the message conveyed by the graphic designer because it distinguishes new trends depicted in a graph from potential trends. Since one application of our message recognition system is to convey the high-level content of information graphics to blind individuals, this differentiation enables us to construct different summaries and avoid misleading the users.

Occasionally the graphic designer wants to emphasize a sudden big change that has occurred. This leads to a Big-Jump and a Big-Fall message category. In such graphs, the graphic designer uses the portion of the graph before a sharp rise or sharp drop to capture the situation prior to the sudden change and the portion of the graph afterwards to convey whether the sudden change was sustained. Thus as discussed in Chapter 7, we differentiate Big-Jump and Big-Fall messages as to whether they are sustained or not sustained. Big-Jump and Big-Fall share similarity with research[99, 83, 78, 77, 73, 100] on detecting interesting patterns in time series data such as gas turbine and neonatal




]

Figure 3.5: A line graph with Point-Correlation intended message

intensive care data. However, the Big-Jump and Big-Fall capture instances where the graphic designer intends to convey a sudden change as opposed to merely using rules to detect spikes in machine-generated data. Figure 3.4 shows examples with Big-Rise and Big-Fall intended messages. Figure 3.4a shows a sudden rise in job cuts between 2000 and 2001; however, this big jump was not sustained. Figure 3.4b conveys a sudden big fall in U.S. airline industry quarterly operating profits between 2000 and 2001.

We have also noticed that some line graphs are intended to correlate events with changes occurring in the data. We refer to this category of messages as Point-Correlation. Although the causality relationship may be unclear, the line graph conveys the message that there is some connection between the events and the changes occurring in the graph. Figure 3.5 illustrates a line graph with a Point-Correlation intended message. It shows the correlation between U.S. military expenditure and the several wars deployed by the U.S. army.

**Summary**

This chapter has presented the categories of messages that are typically conveyed by line graphs. Each of the message categories is defined as a parameterized schema as shown in Table 3.1. Instantiating the parameters with actual values from a graphic produces a possible message that might be what the graphic designer intended to convey. Chapter 4 discusses communicative signals that the graphic designer expects the viewer will use to recognize the graphic's intended message. Chapter 5 then presents the architecture for constructing candidate messages and for exploiting the communicative signals as evidence in choosing among the proposed candidates.

# Chapter 4

# COMMUNICATIVE SIGNALS IN LINE GRAPHS

Just as listeners use evidence to identify the intended meaning of a speaker's utterance, so also must a viewer use evidence to recognize a graphic's intended message. In the case of an utterance, the evidence includes explicit communicative signals such as cue words and intonation as well as the context established by the preceding dialogue and the assumed mutual beliefs of speaker and hearer. Similarly, evidence about the intended message of a line graph includes communicative signals such as annotations of certain points in the graphic as well as features of the line segments themselves. The rest of this chapter discusses the kinds of communicative signals found in simple line graphs; Chapter 7 then discusses how this evidence is extracted from the augmented XML representation of a line graph, and where the communicative signal is captured in a Bayesian network that hypothesizes a graphic's intended message.

## 4.1 Explicit signals by the graphic designer

We hypothesize that if the graphic designer goes to the effort of entering attention-getting devices into a graphic to make one or more of the entities in the graphic particularly salient, then the designer probably intends for these entities to be part of the graphic's intended message. There are several ways in which a graphic designer explicitly makes an entity in a line graph salient, as discussed in the following subsections.

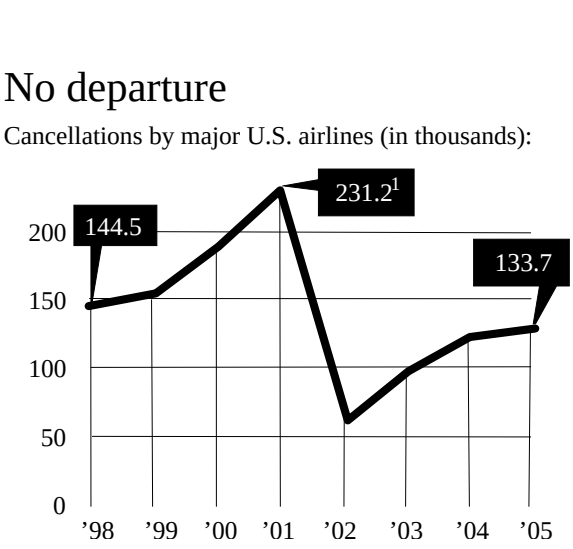



Figure 4.1: Line graph from USA Today with multiple annotations. (This graphic appeared on a slant in its original form.)

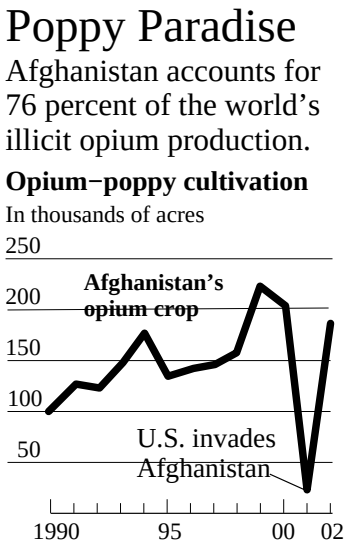



Figure 4.2: A line graph from Newsweek which has an annotation at the lowest point representing an explicit communicative signal

### 4.1.1 Annotations

Elzer[29] found that a bar in a bar chart could be made salient by coloring it differently from the other bars or by annotating it with its value. Similarly the graphic designer may annotate a point on a line graph with a value or a piece of text. This draws attention to that point in the line graph and serves as evidence that the point plays a role in the graphic's intended message. However, in contrast with bar charts, several points may be annotated and become jointly salient. Consider the graphic in Figure 4.1. The high point in the graphic is annotated with its value, as are the two end points. This suggests that these points are particularly important to the graphic's intended message — in terms of our representation, the points might serve as parameters of the graphic's intended message. This provides strong evidence for a Change-Trend-Return(98, rise, 01,fall, 02,rise, 05) message since three of the four annotated points are parameters of the message.

Similarly, consider Figure 4.2. The low point in the graphic is annotated with text, suggesting that it is important to the graphic's message. This annotation might provide evidence for a Big-Fall(00,01) or for a Falling-Trend(00,01) (where the annotation is on the end of the fall), for a Rising-Trend(01,02) (where the annotation is on the start of the rise), for a Change-Trend-Return (where the annotation is on the point where the return begins), or perhaps for a Change-Trend (where the annotation is on the point at which the trend changes). Alternatively, since the annotation is text, it could provide evidence for a Point-Correlation message.

### 4.1.2 Nouns in the caption/description [1]

Elzer[28] found that a bar in a bar chart becomes salient when a noun in a caption matches the label of the bar. Similarly a point in a line graph can become salient by virtue of its being referenced by a noun in the caption. However, in contrast with bar charts, this can occur in one of two ways: by the caption referring to the point's x-axis value or to the point's y-axis value, although the latter occurs less often. For example, if the caption on the graphic in Figure 4.2 were *"Poppies Missing in 01"*, the reference to the year "01" would lend salience to the low point in the graphic even if it were not annotated. And in Figure 4.3, the number one-seventh in the caption makes the last point of the line graph salient because the y-value of the last point is approximately $1/7$ as referred to by the caption.

### 4.1.3 Other signals in the caption/description

As shown by Corio and LaPalme[18] and by our own corpus study, captions are often very general and do not capture a graphic's intended message. For example, the caption on the graphic in Figure 4.2 fails to capture its message that there was a sudden

[1]The description is a piece of text following the caption of an information graphic to complement its content. For example, the sentence *"Afghanistan accounts for 76 percent of the world's illicit opium production"* is the description in Figure 4.2.

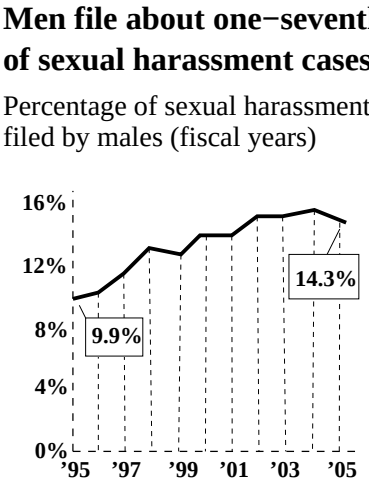



Figure 4.3: The last point is salient because its y axis value is referred by the text in the Caption

big fall (that was not sustained) in Afghanistan opium production. Moreover, even when a caption conveys some of the graphic's message, it is often ill-formed or requires extensive world knowledge to understand. However, as in Elzer's work[28] on simple bar charts, captions often contain simple signals that help identify the graphic's message. Elzer[28] found that verbs and adjectives in a caption suggest the general category of message. We have found the same to be true for line graphs. For example, the word *"decline"* in the caption of Figure 4.4 suggests a Falling-Trend message or perhaps a Change-Trend message where the trends change from rising to falling. In another example shown in Figure 4.5, the word "bumpy" suggests that there is a change in the line graph.

## 4.2 Salient features inherent in a line graph

Certain parts of a graphic become salient without any effort on the part of the graphic designer. For example, a viewer's attention will be drawn to a sudden large rise or fall in a line graph — one that not only has a large absolute slope but also represents a large change in value relative to the range of values depicted in the graph, such as in Figure 4.6. Similarly, a viewer will be interested in the segment at the end of a line graph both because it captures the end of the quantitative changes being depicted and because

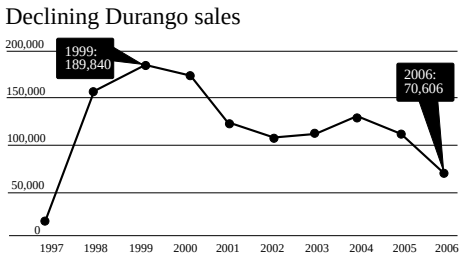



Figure 4.4: Line graph from USA Today with a helpful word in the caption.

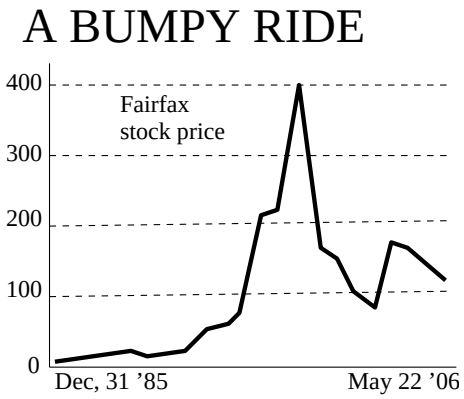



Figure 4.5: A line graph where a word in the Caption is used as a communicative signal.

the end of the line graph may display the most recent data that has an affect on real life at the moment that the line graph is published. Although no specific effort is required by the graph designer, we posit that it is mutually believed by both graph designer and viewer that such pieces of the graphic will be salient.

### 4.3 Evidence from other features of the graphic

Several features of the sequence of points in a graph covered by a suggested message also provide evidence for or against that proposed message being the intended message of the graphic. The graphic designer presumably had a reason for including all of the points in a line graph. Thus the fraction of a line graph covered by the portion of a graph comprising a suggested message serves as evidence about whether that was the graphic designer's intended message — presumably, messages that cover much of the line graph are more likely to be the designer's intended message. (However, the intended message need not cover the entire graphic. For example, it appears that when conveying a Rising-Trend, the graphic designer sometimes includes a small number of points prior to the start of the trend in order to keep the viewer from inferring that the rise might have started at earlier points not depicted in the graphic.) As discussed earlier, viewers are naturally interested in the end of the line graph, particularly if it reflects recent events. However,

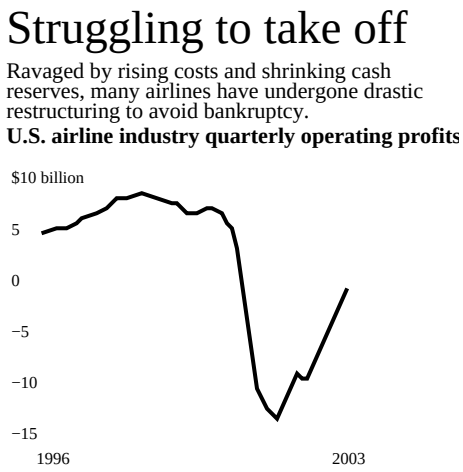



Figure 4.6: A line graph with a sudden large change in value relative to the range of values depicted in the graph.

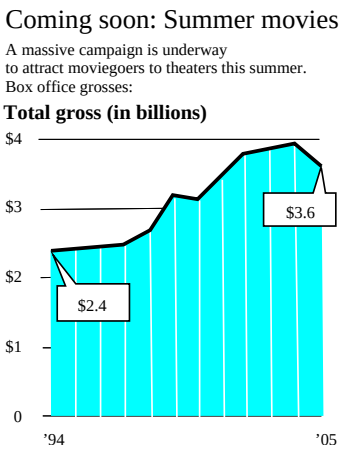



Figure 4.7: A line graph in which the small last piece may draw the attention of the reader.

a segment at the end may be short and only suggest the possibility of a new trend, rather than capturing an actual trend in the data, such as in Figure 4.7. Thus the relative width of the last segment serves as evidence for or against messages such as Change-Trend and Contrast-Trend-Last-Segment.

## 4.4 Summary

This chapter has identified the kind of communicative signals that appear in line graphs. Chapter 7 will discuss how to extract them from a line graph. An overview of our architecture for utilizing these communicative signals to hypothesize the intended message of a line graph is presented in Chapter 5.

# Chapter 5

# SYSTEM ARCHITECTURE

To recognize the intended message of a line graph, we have to know what kinds of intended messages could be conveyed by the line graph designer. Chapter 3 discussed the kind of messages conveyed by line graphs and represented them as message categories with parameters. Since a graphic designer for popular media is not only showing data to the reader as in scientific articles but is also trying to convey a message, he/she will use multiple clues(i.e. communicative signals) to help the reader achieve his/her communicative goal. The kinds of communicative signals present in line graphs was described in Chapter 4. This chapter discusses the motivation for our overall approach and presents our system architecture for inferring the intended message of a line graph.

## 5.1 Motivation for the overall approach

In Chapter 2 on related work, we discussed how the line graph designer uses multiple visual features in designing a line graph and then the human reader plugs those visual features into a graph schema to comprehend the line graph. Although we are not trying to simulate human graph comprehension, we can relate our work to Pinker's[71] concepts of visual descriptions and graph schema, and view our work as constructing a graph schema which utilizes the communicative signals (which can be regarded as visual descriptions) to identify the graph's intended message.

In our research, we assume that line graphs from popular media shouldn't require a specific graph schema since the audience consists of normal people without extensive

domain knowledge. So our goal is to construct a general graph schema without using domain knowledge such as stock market knowledge or geographical knowledge. The graph schema discussed in [71] is an automata with vertex representing the visual features and the directional edges representing the relationship between visual features. We have also chosen a graphical model[48, 70] in which the communicative signals are connected to possible candidate messages but we want to add probabilistic inference to the graphical model; thus we have selected a Bayesian network[48, 70, 22, 3] as our representation and inference model.

To let the inference model determine the final intended message of a line graph, we need to provide it with a set of intended message candidates, each associated with the evidence provided by their corresponding communicative signals, to do probabilistic reasoning. These different kinds of communicative signals provide viewers with clues to infer the intended message of a line graph. The Bayesian network inference model uses the a priori distribution and likelihood of evidence to update the posterior probability of the intended message for the line graph. After the belief update, the Bayesian network chooses among multiple candidate messages, where a candidate message is an intended message category with instantiated parameters which can potentially be the true intended message of a line graph. The intended message might be ambiguous and the probabilistic inference within the Bayesian network arbitrates among the possible candidate messages by giving us a probabilistic confidence of each intended message instead of casting a hard classification of the line graph to a single category. The Bayesian network learns the model (the a priori distribution and the conditional probability tables) from real data instead of being manipulated by researchers based on domain knowledge so that the graph schema does not reflect human-generated rules and the bias is minimized. Learning from training data from different domains also provides a general graph schema which doesn't rely on domain knowledge.

The Bayesian network must be provided with a set of candidate messages from

which to choose, thus a mechanism for suggesting candidate messages is required. For line graphs, the suggestion of intended message candidates differs from other kinds of information graphics. A line graph is continuous but we need to work with discrete entities. Even if we sample the line graph to get a set of discrete data points, it is impossible to consider every possible data point as an instantiation of a parameter in the message categories. For example, if there are $n$ sampled data points for a line graph, the simplest Rising-Trend intended message (with two parameters) will have $n(n-1)/2$ possible parameter pairs, so there will be $n(n-1)/2$ candidate messages for Rising-Trend category only. According to the grouping principle discussed in [49] on the limitation of the human brain, the series of data points tend to be grouped into a smaller set of segments by the human reader. Although our research is not attempting to simulate the way humans comprehend line graphs, the human perception and comprehension process indeed suggests the following:

1. The curved line should be segmented into a few visually distinguishable trends. This is supported by the grouping principle discussed in [49].

2. An annotation in the graphic should not only be matched with a single data point but also to the beginning or ending of a trend segment if the end-points of a trend segment are close to the annotation. Associating the annotations with trends is supported by the psychological fact that people do not like to expend effort; associating annotations with single data points requires the reader to spend more effort on judging the influence of the annotation on other adjacent data points.[49]

Thus if a line graph is segmented into a small set of visually distinguishable trends, candidate messages (suggestions) can be proposed based on the segmented visually distinguishable trends. Each intended message candidate has a set of communicative signals so that the Bayesian network can not only hypothesize the correct intended message from candidates but also has the communicative signals of each candidate as evidence for or against the candidate in the inference process.

## 5.2 System architecture

Figure 5.1 shows the overall architecture of our system for recognizing the intended message of a line graph. A Visual Extraction Module[14] is responsible for analyzing the graphic and producing a XML representation. The XML representation is a complete specification of the graphic, including the following:

1. A sampling of the data points, thereby discretizing a continuous line graph into a set of sampled data points.

2. The annotations on the line graph.

3. The full caption and description: the caption is the title of the line graph, and the description consists of the sentences that expand on the caption. Figure 5.2a and Figure 5.2b both have a caption and a description. The captions are *"Coming soon: Summer movies"* and *"Poppy paradise"*, and the descriptions are *"A massive campaign is underway to attract moviegoers to theaters this summer."* and *"Afghanistan accounts for 76 percent of the world's illicit opium production"* respectively.

4. Text-in-graphic, which is a piece of text appearing in the graphic area. The words "Afghanistan's opium crop" inside Figure 5.2b is a text-in-graphic element.

5. Axis labels, tick-marks on both axes, etc.

After the Visual Extraction Module generates the XML containing the line graph elements and the raw sample points, the Caption Tagging Module[28] is responsible for extracting communicative signals from the caption/description and producing an augmented XML representation that includes this information. It is described further in Chapter 7.

The Message Recognition Module takes as input the augmented XML representation of a graphic and produces as output a logical representation of the graphic's intended message. The Message Recognition Module includes several submodules: the Graph

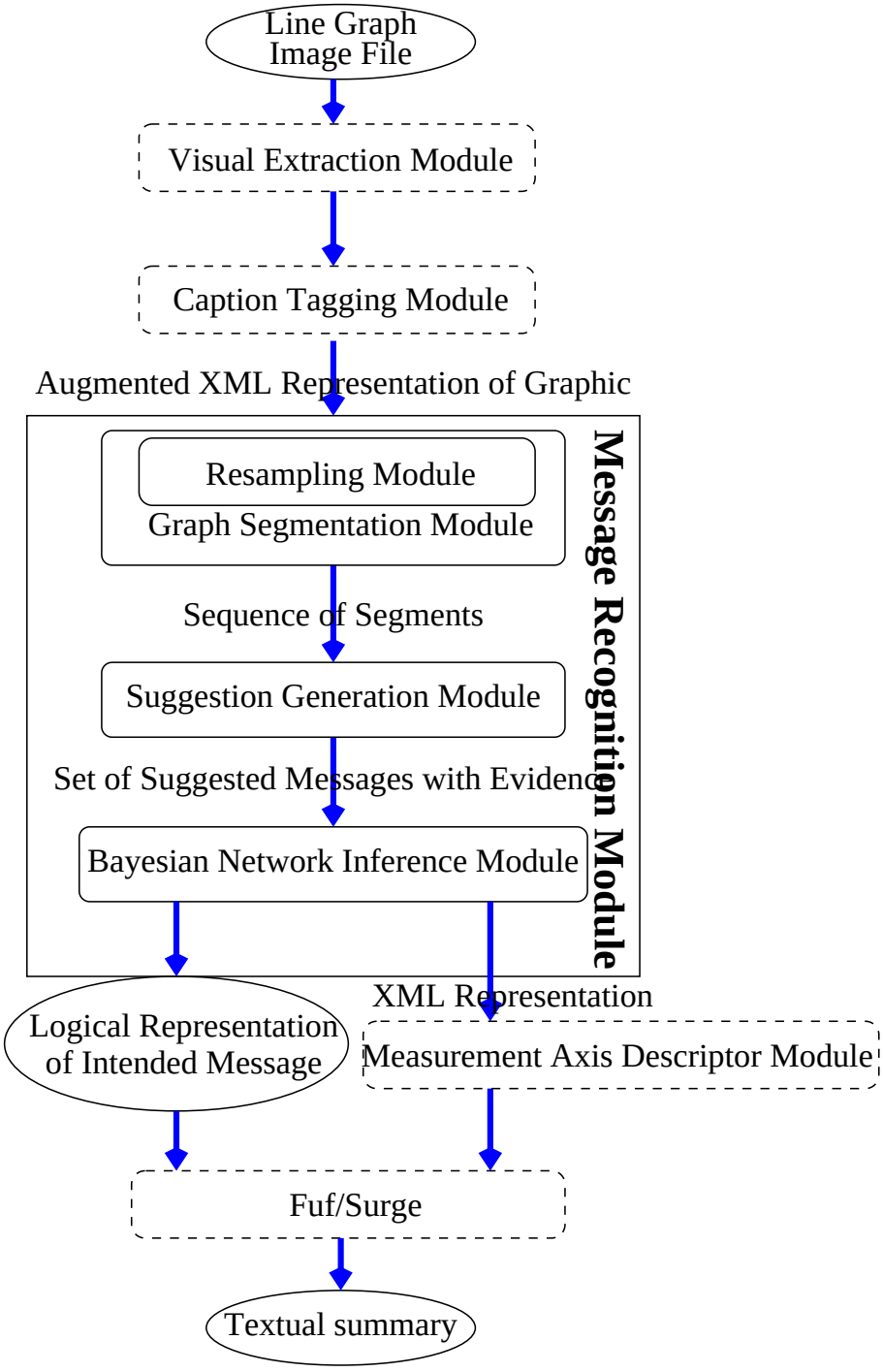
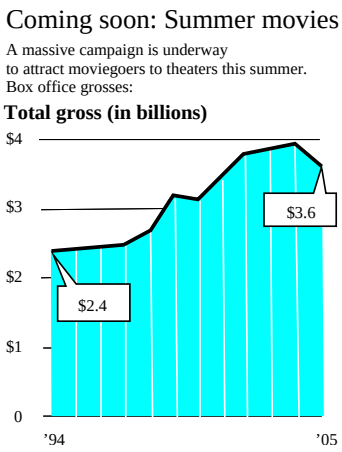
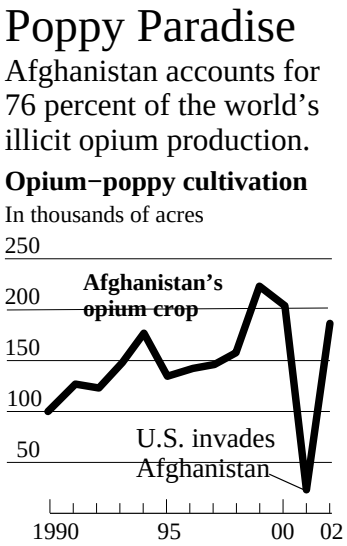



Figure 5.1: Overall Architecture




(a) Line graph having Caption and Description

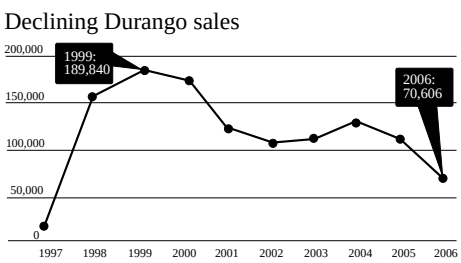



(b) Line graph having Caption, Description and Text-in-graphic

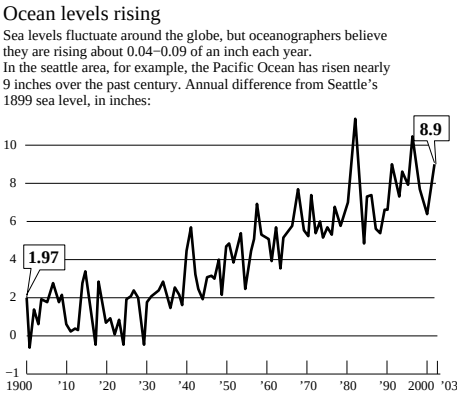
Figure 5.2: Two example line graphs showing the Caption, Description and Text-In-Graphic

Segmentation Module which breaks the line graph into a set of visually distinguishable trends, the Suggestion Generation Module which uses the results from the Graph Segmentation Module to generate a set of candidate messages, and a Bayesian Network Inference Module which collects the evidence (present as communicative signals in the graph) for or against each candidate, and enters the candidates and collected evidence into a Bayesian network which identifies the candidate message with the largest posterior probability as the recognized intended message. The Graph Segmentation Module is elaborated in Chapter 6. The Suggestion Generation Module and Bayesian Network Inference module will be discussed in Chapter 7.

After a logical representation of the intended message is generated, FUF/SURGE is used to generate a natural language sentence conveying the line graph's intended message. However, the logical representation produced by the Bayesian network does not include a referent for the dependent axis and this is needed in order to generate a coherent natural language sentence. Unfortunately, line graphs often fail to explicitly label what is




(a) Line graph having Caption and Description




(b) Line graph having Caption, Description and Text-In-Graphic

Figure 5.3: Two example line graphs whose measurement axis descriptor should be extracted bythe Measurement Axis Descriptor Module

being measured on the dependent axis. For example, the dependent axes on the graphs in Figure 5.3a and Figure 5.3b are not labelled, but the referents "Durango sales" and "annual difference from Seattle's 1899 sea level, in inches" respectively are needed to generate an English realization of the intended message. Thus a module named Measurement Axis Descriptor Module is needed to identify this referent from other information in the graphic. It is discussed further in Chapter 9.

The Visual Extraction Module was developed by Daniel Chester[14]. The Measurement Axis Descriptor Module was designed by Seniz Demir. My research is encapsulated in the Message Recognition Module which is shown in rectangles with solid lines in Figure 5.1.

## 5.3 Summary

This chapter has presented an overview of our framework for recognizing the intended messages of line graphs. Using Pinker's[71] concepts of visual descriptions and graph schema, our approach is to construct a general graph schema to utilize visually distinguishable trends and communicative signals which can be regarded as visual descriptions. Our research is not attempting to simulate the way humans comprehend line

graphs. However, the human perception and comprehension process supports our design of the Message Recognition Module which contains multiple sub-modules. The details of each module will be discussed in the following chapters.

# Chapter 6

# SEGMENTING A LINE GRAPH INTO VISUALLY DISTINGUISHABLE TRENDS

In the related work in Chapter 2, we discussed how the line graph designer uses multiple visual features in a line graph and these are recognized by human readers and plugged into the general graph schema to comprehend the line graph. According to the grouping principle discussed in [49] on the limitation of the human brain, the series of data points tend to be grouped into a smaller set of segments by the human viewer who has difficulty holding more than four entities in mind at once[49]. There is a also psychological principle that people do not like to expend effort and often will not bother to do so, particularly if they are not sure in advance that the effort will be rewarded[49].

Our research is not attempting to simulate the way humans comprehend line graphs. However, the human perception and comprehension process indeed supports our hypothesis that the series of data points should be segmented into visually distinguishable trends rather than working with the large set of data points connected by short line segments. As an example, we see that Figure 6.1 shows two visually distinguishable trends for ocean levels — a relatively stable trend from 1900 to 1930 and a rising trend from 1930 to 2003 (both with high variance).

This chapter presents our model for segmenting a line graph into a set of visually apparent trends. The model is constructed by a support vector machine that takes into account both local and global attributes. The advantage of using machine learning to produce the graph segmentation model is that a machine learning algorithm can consider

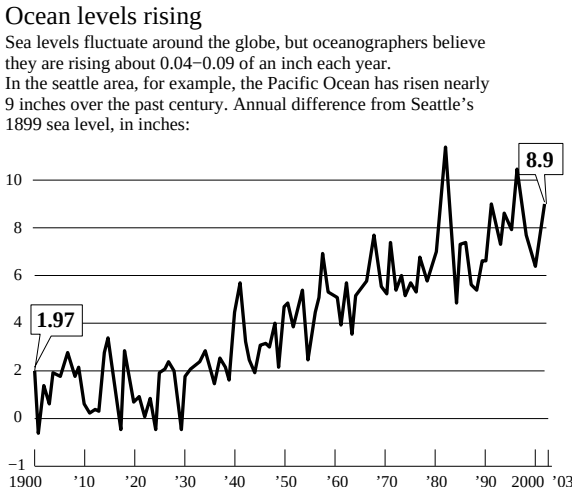



Figure 6.1: A jagged line graph

a variety of candidate attributes and emphasize those that are important in producing a segmentation that captures trends which are visually apparent to humans, as opposed to using an algorithm that comes from the perspective of error minimization. To our knowledge, our work is the first approach to graph segmentation that 1) captures trends that are visually apparent to humans, 2) uses both global and local information, and 3) uses machine learning to produce a learned graph segmentation model.

This chapter is organized as follows: Section 6.1 discusses related work which is primarily on time series segmentation done by researchers from computer science or mathematics. Section 6.2 describes our approach to building a model of graph segmentation. Section 6.3 presents several examples of segmentations produced by our system, and Section 6.4 presents our evaluation experiments, including a cross-validation of our decision module that determines whether to split a segment, a comparative evaluation with another method, and an evaluation of the quality of our segmentations. We work with hand-coded XML representations of line graphs which contain a relatively uniform sampling of points in the line graph, including change points. This is because the Visual Extraction Module currently can handle only a few of the line graphs in our corpus. Appendix A presents a Resampling Module that addresses the issue of converting the data

points provided by the VEM into the set of sampled data points needed for the statistical tests used in the Graph Segmentation Module.

## 6.1 Related work

Our line graph segmentation task is related to research on time series segmentation. One major group of researchers[41, 46, 39, 88, 87, 43] use the top-down, bottom-up or sliding window approaches to splitting a time series into segments. The top-down approach inserts splitting points (locations specified for dividing one segment into two sub-segments) into the segment recursively until the constraint within each segment or the constraints on the whole set of segments are satisfied. The bottom-up approach starts with each pair of adjacent data points as a segment and each non-boundary data point as a splitting point, and keeps merging segments by removing splitting points, until some criterion is reached. The sliding window method does not view the whole data set at once but instead moves a fixed size window from one side to the other side. It accepts a segment whenever a continuous series of data points within the window satisfies a criteria, such as reaching an error upper bound, etc. It is appropriate for online segmentation where an unknown data stream is processed as input.

Most of these projects focused on splitting a given time series into a number of segments by finding the piecewise linear approximation or piecewise aggregate approximation which provides the smallest total error or conforms to a maximum error bound within each segment. Piecewise linear approximation uses a regression line to represent a segment and piecewise aggregate approximation uses the horizontal segment whose y-value is the mean value of the data points within the segment to represent it. These research efforts either ask for an a priori fixed number of segments or place a fixed upper bound on errors, either on each individual segment or on the whole series of segments. These thresholds require prior knowledge about the time series data and can limit the usage of a method. Although some research[43, 90] also suggested using AIC (Akaike Information Criterion)or BIC (Bayesian Information Criterion) to trade off between the

total amount of error and the total number of splitting points or the degrees of freedom (in the regression formula), the remaining problem of where to split is still an active research issue. So for the general time series segmentation algorithm based on error reduction, the two problems are where to split and when to stop. Our algorithm uses a splitting module and a decision module to handle the two problems, and the decision module is a learned model from human judgement data, instead of using a fixed error bound or parameters (such as the number of segments provided as result).

Besides the error reduction methods, another time series segmentation algorithm[79] applies the formalism of knowledge-based temporal abstraction to the sliding window according to a priori rules extracted from knowledge about the time series data. Another kind of similarity based time series segmentation research[52, 15] matches the time series data to some predefined patterns, and thus the predefined pattern categories have a large influence on the final result. The a priori knowledge dependent methods are not easy to generalize to data sets from different domains.

As opposed to the above piecewise approximation approach, another set of time series segmentation research focuses on detecting change points or anomalies in the data. The ARIMA (autoregressive integrated moving average)[13] or ARMA (autoregressive moving average)[95] models fit a formula to the existing data points and predict the range of the incoming data point. Some other regression models[63] fit the time series data using regression methods and calculate the confidence interval of the incoming data point. Change points are identified at those data points outside the predicted upper/lower bound or the defined confidence interval. These methods face the problem of determining a priori how many past data points should be used in their prediction model. Furthermore, the change point detection procedure is very susceptible to the chosen model and distribution. Another set of change point detection methods, such as maximum vertical distance[59], also require a priori knowledge to define the threshold used in the model, and different thresholds will yield much different results. Similar to our project, some time series

segmentation methods[36, 55, 68] rely on statistical tests. However, they use only one or two statistical tests to determine if a segment should be split into subsegments.

These research efforts differ from our work in several ways. They are not concerned with extracting visually identifiable trends but are instead concerned with segmenting based on error minimization, or with pattern detection, prediction, or anomaly detection. In addition, they use only one or two statistical tests or a limited amount of information calculated from one criterion; although they are able to describe the segment under consideration for their particular usage, they are not able to fully capture the characteristics of the segment. In our research, the decision module uses a machine learning framework to combine a wide variety of features and construct a learned model.

## 6.2 Problem formulation and algorithms

### 6.2.1 General framework

Given a series of sampled data points from a line graph, we need to segment this data set into one or several sequences where each sequence of sample points can be represented by either piecewise linear interpolation or piecewise linear approximation[46] to show a visually distinguishable trend.

The data in a line graph can be discretized into a set $P$ of two dimensional sample points, $P = \{P_k | k = 1, ..., n\}$, with the $i$th data point $P_i$ having $x_{P_i}$ and $y_{P_i}$ as coordinates. The problem of segmenting the line graph into a sequence of visually distinguishable trends can be described as constructing a set of ordered pairs
$Q = \{< Q_{k,1}, Q_{k,2} >, k = 1, ..., m\}$, where $m < n$, $Q_{k,i} \in P$, $Q_{1,1} = P_1$ and $Q_{m,2} = P_n$. Each pair $< Q_{k,1}, Q_{k,2} >$ represents a trend in the line graph, with $x_{Q_{k,1}} < x_{Q_{k,2}}$ and $x_{Q_{k,2}} = x_{Q_{k+1,1}}$, i.e. the end point of the first trend segment is the starting point of the second trend segment.

As in [46], the trend can be represented by either piecewise linear interpolation, which connects $Q_{k,1}$ and $Q_{k,2}$ with a straight line, or by piecewise linear approximation,

which calculates a least squared regression line using all data points $P_i$ in $P$ such that $x_{Q_{k,1}} \leq x_{P_i} \leq x_{Q_{k,2}}$.

The future application of our research requires that we use a method that has three characteristics:

1. A fast algorithm so that it can be used in a message recognition system that produces a natural language description of the high-level content of a given line graph in real time, thus providing blind users with efficient alternative access to information graphics.

2. Some line graphs can be very smooth, but others can be very jagged with large variance, as in Figure 6.1. Rather than having a set number of segments or threshold of error, our algorithm must deal with a wide variety of line graphs and determine the number of segments as it produces the segmentation for a particular graphic.

3. The existing algorithms for time series segmentation are mainly based on error reduction methods. Since our goal is to identify a segmentation that captures human perception of visually apparent trends, we must use machine learning to consider a variety of different attributes and produce a learned model that emphasizes the most important attributes in identifying visually distinguishable trends.

We have chosen a top-down approach for our segmentation task[46]. An advantage of the top-down approach for our purpose is, as the segmentation moves from the whole graph to individual segments, it is possible to record global information about the larger graphic and pass it for consideration when analyzing the subsegments and deciding whether to divide them further.

Our segmentation algorithm is a recursive algorithm that starts from the whole line graph as one segment. Given a segment as input, the decision module makes a split/no-split decision. If a split decision is made, the splitting module will determine the splitting point, split this segment into two subsegments, and call the decision module on each new

subsegment. Recursion stops when the decision module makes a no-split decision on a segment. The following algorithm shows our mechanism in pseudo-code:

**Algorithm 1** segmentation algorithm using splitting module and decision module

$Q \Leftarrow < P_1, P_n >$
**repeat**
  Let $< Q_{k,1}, Q_{k,2} >$ be the first unprocessed ordered pair of $Q$
  $Q \Leftarrow Q - < Q_{k,1}, Q_{k,2} >$
  **if** DecisionModule($< Q_{k,1}, Q_{k,2} >$)=split **then**
    $P_s$ =SplittingModule($< Q_{k,1}, Q_{k,2} >$)
    $Q \Leftarrow Q \bigcup \{< Q_{k,1}, P_s >, < P_s, Q_{k,2} >\}$
  **else**
    $Q \Leftarrow Q \bigcup \{< Q_{k,1}, Q_{k,2} >\}$
  **end if**
**until** all pairs in $Q$ are processed by DecisionModule
Return $Q$ as the segmentation

The splitting module and decision module are covered in Section 6.2.2 and Section 6.2.3 respectively.

### 6.2.2 Splitting module

The splitting module is responsible for selecting the splitting point for each segment once the decision module determines that the segment should be split. Fu-lai Chung et. al.[52] introduced a simple method which uses the PIP (perceptual important point) as the splitting point in a segment. The idea can be simply described as finding the point which has the largest perpendicular distance from the straight line which connects the two endpoints of the segment. The formula can be represented as:

$$\arg\max_{\mathbf{k}}(y_k - \begin{bmatrix} 1 & x_k \end{bmatrix} \begin{bmatrix} 1 & x_i \\ 1 & x_j \end{bmatrix}^{-1} \begin{bmatrix} y_i \\ y_j \end{bmatrix})$$

Choosing the PIP as the splitting point is only complexity $O(n)$, as opposed to choosing a splitting point that minimizes the sum of squared errors which would be $O(n^2)$. However, human perception is sometimes more sensitive to the maximum or

minimum point than to the PIP. If two points are close to one another, with one being the PIP $P_{PIP}$ and the other being the maximum/minimum point $P_M$, we hypothesize that it might be better to select the maximum/minimum point as the end of one trend segment and the start of the next trend segment. For example in Figure 6.2, the PIP point and maximum/minimum point are both circled, with the PIP point on the left and maximum/minimum point on the right. In this case, if the decision module made a split decision, the maximum/minimum point appears to be a better choice for splitting the line graph than the PIP point. Thus we consider both the PIP and the maximum/minimum points as candidate splitting points. To choose between them, we examine how much one data point stands out against the other with respect to its own direction (perpendicular or vertical), by comparing 1) the difference $D_p$ in their perpendicular distances from the straight line connecting the two end points of the segment against 2) the difference $D_v$ in their vertical locations (or y-values), as shown in Figure 6.3. If $D_p \geq D_v$, we choose $P_{PIP}$ as the split point; otherwise, we choose $P_M$.

### 6.2.3 Decision module

The Decision Module is responsible for analyzing a segment and making a decision about whether it should be split into two subsegments. Since the output of the decision module is a binary value, it can be reduced to a binary classification problem. In our project, we use 18 local and global attributes and a support vector machine with SMO (Sequential Minimal Optimization)[72]. We explored decision tree learning since the resultant tree is easy to understand and analyze, but as the number of features increases, a decision tree may encounter an over-fitting problem because the decision branches increase exponentially as the number of features increase. A support vector machine can overcome this overfitting problem as the feature set increases[5, 85], so it is better for possible future research with an enlarged feature set.

We build the feature base from both local and global information. As opposed to other segmentation algorithms which only consider local information obtained solely

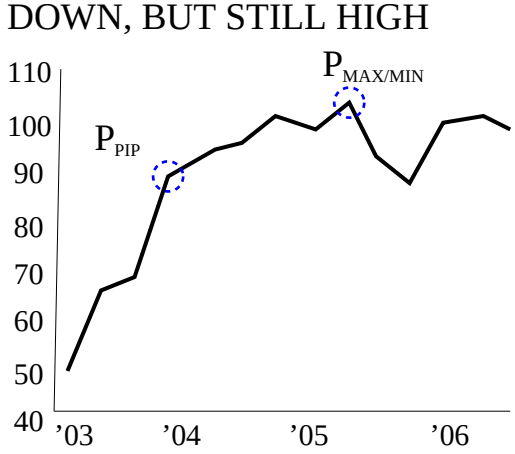



Figure 6.2: The PIP point and the maximu/minimum point are circled in this line graph, with the PIP point on the left and the maximum/minimum point on the right. In this case, choosing the maximum/minimum point to split the graph is better than choosing the PIP point

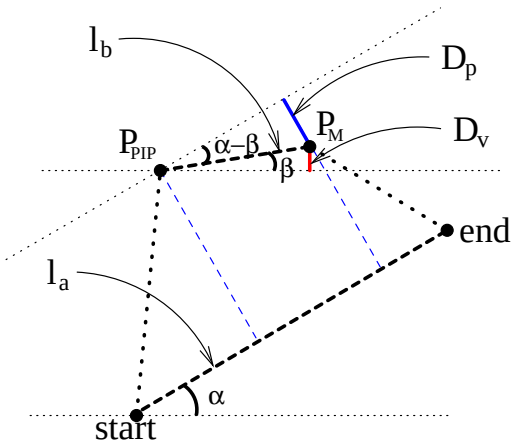



Figure 6.3: Relationship between $P_{PIP}$ and $P_M$

from the segment under consideration, we also take advantage of global information obtained from outside the segment (such as the relative length of the segment with respect to the whole line graph). Section 6.2.4 describes the local features, Section 6.2.5 describes the global features, and Section 6.2.6 describes our use of a support vector machine.

### 6.2.4 Local features

The local feature base is composed of various statistical tests on the segment and other attributes which represent characteristics of the segment. The following discusses all of the local features that are considered by the SVM in building the decision module for graph segmentation.

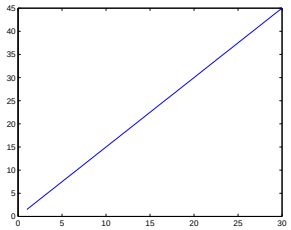



Figure 6.4: Correlation coefficient=1, should not be split

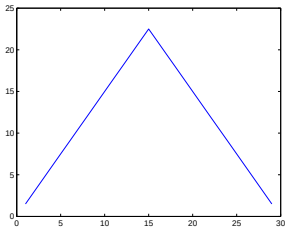



Figure 6.5: Correlation coefficient=0, should be split

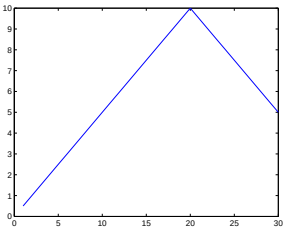



Figure 6.6: Correlation coefficient=0.706, should be split

#### 6.2.4.1 Correlation coefficient

A trend can be viewed as a linear relation between the $X$ and $Y$ variables. The Pearson product-moment correlation coefficient measures the tendency of the dependent variable to have a rising or falling linear relationship with the independent variable. It is obtained by dividing the covariance of two random variables $X$,$Y$ by the product of their standard deviation.

$$r_{XY} = \frac{n\sum x_i\, y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\,\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

The correlation is between $-1$ and $1$, where $1$ means an increasing linear relationship and -1 means a decreasing linear relationship. The closer the coefficient is to either 1 or -1, the stronger the correlation between the variables. We use the absolute value of the correlation coefficient as a feature in our classifier. Because the correlation coefficient measures the strength of the linear dependence between two variables, we hypothesize that the strong linear segment shouldn't be split as in Figure 6.4 and that a low correlation coefficient suggests that the segment is not linear and should be split as in Figure 6.5 and Figure 6.6.

#### 6.2.4.2 Q-test and F-test

Although the correlation coefficient is useful in detecting when a segment should be viewed as a single trend (and thus not split further), it is not sufficient by itself. A

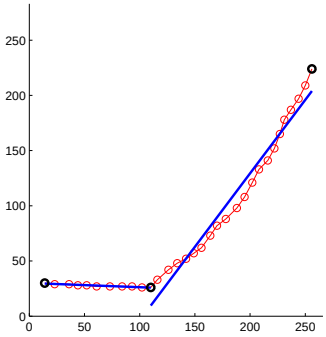



Figure 6.7: An example line graph with high correlation coefficient but which should be split into two subsegments, as indicated by the dark circles. The light colored circles are the sampled data points, the three dark circles are the splitting points, and the solid lines are the regression lines for the two subsegments.

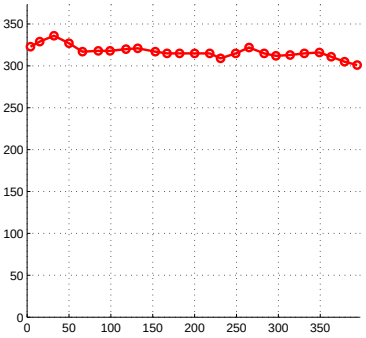



Figure 6.8: Graph with low correlation coefficient, but which should be treated as a single trend

flat portion of a line graph may be followed by a very steep rise, resulting in a high correlation coefficient even though the graph should be split into two segments, as in Figure 6.7. Similarly, a relatively flat segment, such as the line graph in Figure 6.8, will have a low correlation coefficient, even though it should not be split into subsegments.

To address this, we make use of the Q-test[74] and F-test[92, 2] which are measures of change point detection; both test whether a two-segment regression is significantly different from a one-segment regression based on the differences in their respective residuals. The null hypothesis is that no change point has occurred so the two regression models are equal, suggesting that the segment need not be split further into subsegments. The Q-test is specifically designed for this purpose. But as analyzed in [74], the Q-test has less power when the change point is closer to the two endpoints. And according to [27], the Q-test is sensitive to the change in sample size – the larger the sample size, the better its performance. On the other hand, the F-test was designed as a general model fitting test

but has been adapted to the two-phase regression problem. The F-test may compensate for problems with the Q-test since it empirically works better when the change point is towards one of the two endpoints. Because the Q-test and F-test are both calculated from residuals, they can be calculated together, thereby reducing the computational complexity, although they have different statistics and critical values.

Both the Q-test and the F-test postulate the existence of two relationships in a given segment consisting of a sequence of data points $(x_i, y_i), i = 1, 2, ..., n$. These two relationships can be written as:

$$y_i = \begin{cases} a_1 x_i + b_1 + u_1 & \text{if } i \leq k \\ a_2 x_i + b_2 + u_2 & \text{if } i > k \end{cases} \tag{6.1}$$

where $u_1$ and $u_2$ are normally and independently distributed error terms with mean zero and standard deviations $\sigma_1$ and $\sigma_2$. The null hypothesis for both tests is $H_0 : a_1 = a_2$, $b_1 = b_2$, $\sigma_1 = \sigma_2$ against $H_1 : a_1 \neq a_2$ or $b_1 \neq b_2$ or $\sigma_1 \neq \sigma_2$. Figure 6.7 plots the two relationships with two solid lines where the circled data point in the middle represents the $k$ in Equation 6.1. Accepting the null hypothesis means that a one-phase regression model better captures the segment, and the segment should not be split.

The Q-test analyzes the likelihoods of the first $k$ data points and the following $n - k$ data points from a Gaussian distribution and takes the logarithm of the likelihood ratio $\lambda$:

$$\lambda = \frac{l(k)}{l(n)} = \frac{\hat{\sigma_1}^k \, \hat{\sigma_2}^{n-k}}{\hat{\sigma}^n}$$

where $\hat{\sigma_1}$ and $\hat{\sigma_2}$ are the estimates of the standard errors of the two regression lines, and $\hat{\sigma}$ is the estimate of the standard error of the overall regression line for all data points in the segment. According to Quandt[75], the statistic $-log(\lambda)$ follows the distribution table listed in [75].

The F-test statistic is computed as

$$F = \frac{(RSSL - RSS)/2}{RSS/(n-4)}$$

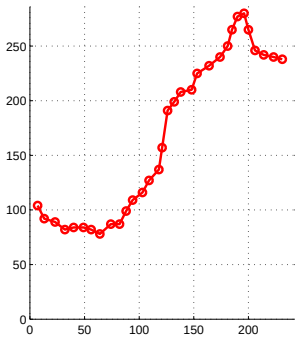



Figure 6.9: Line graph with three trends in it, sampled from Business Week

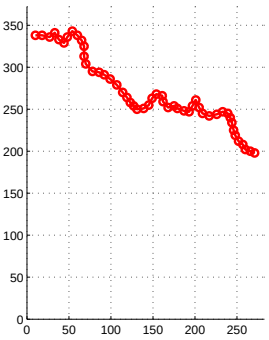



Figure 6.10: Line graph of falling trend, sampled from USA Today

where RSSL is the residual sum of squares of the overall regression line, and RSS is the residual sum of squares of the two-phase piecewise regression lines. The value $F$ here is distributed as an F-distribution with $(2, n-4)$ degrees of freedom as given in [2].

For each sample point $P_k$ in a segment, where $1 < k < n-1$ and $n$ is the total number of data points in the segment, we build the two-phase linear regression models for the data points from $P_1$ to $P_k$, and from $P_{k+1}$ to $P_n$ respectively. We consider three significance levels $\alpha = 0.05$, $\alpha = 0.01$, and $\alpha = 0.005$ for the Q-test, and two significance levels $\alpha = 0.1$ and $\alpha = 0.05$ for the F-test in the feature base. Thus we consider a total of five attributes based on the two tests. If a statistic calculated from any $k$ where $1 < k < n-1$ generates a significant result corresponding to one of the significance levels of the Q-test or the F-test, we set the value of the corresponding attribute to 1; otherwise, if no $1 < k < n-1$ in the given segment makes the statistic significant for the corresponding significance level, the attribute is set to 0.

#### 6.2.4.3 Runs test

Although the two-phase Q-test and F-test may help address the problem of recognizing segments that represent a sequence of two trends, they can give a wrong prediction

in some situations where the segment consists of more than two trends, such as in Figure 6.9. However, computational complexity prevents us from calculating more than a two phase F-test to fit the data points since $O(n^d)$ combination of points would need to be considered for a $d-1$ phase F-test. Failing to reject the null hypothesis for the one-phase regression model doesn't mean the one-phase regression line necessarily fits the data points well since the data points may be better captured by a $k$-phase regression model where $k > 2$. Therefore, we need a more general statistic to test the goodness of fit between the piecewise linear regression model and the data points.

We make recourse to the Runs Test[4]. The Runs Test detects if a model fits the data points well. For each data point, we calculate its residual from the regression line and categorize it as $+1$ or $-1$, according to whether the residual is positive or negative. Then the number of *runs* is calculated, where a *run* is a continuous sequence of residuals which belong to the same category, such as consecutive $+1$ or $-1$. If $N_+$ is the number of positive residual points and $N_-$ is the number of negative residual points, the mean and standard deviation of the number of runs suggested by the data points are approximated as

$$R_{mean} = \frac{2\,N_+\,N_-}{N_+\,+\,N_-} + 1$$

$$R_{SD} = \sqrt{\frac{2\,N_+\,N_-\,(2\,N_+\,N_-\,-\,N_+\,-\,N_-)}{(N_+\,+\,N_-)^2\,(N_+\,+\,N_-\,-1)}}$$

If the number of runs computed from the data points is sufficiently close to $R_{mean} \pm R_{SD}$, the residual is probably a reasonable approximation of the error from the regression, and this regression model may be regarded as a good fit to the data points.

We use the Runs Test to check how well the least squared linear regression for a given segment fits the data points in the segment. If the actual number of *runs* $R$ is larger than $R_{mean} - R_{SD}$, it suggests the least squared linear regression line is a good fit to the segment. We include five features from the Runs Test: the result of the Runs Test, the

actual runs $R$, mean runs $R_{mean}$, standard deviation of runs $R_{SD}$ for a segment, and the ratio difference between actual runs and mean runs calculated as $|R - R_{mean}|/R_{mean}$.

Although the Runs Test appears powerful in suggesting whether a segment should be split further, it alone is insufficient. The Runs Test only uses the sign of the residual, not its value. It may suggest that the line graph in Figure 6.10 should be split, rather than viewing it as a single falling trend. However, other attributes, such as the correlation coefficient discussed earlier, will suggest otherwise.

#### 6.2.4.4 Outlier detection

A line graph may have one or more points that significantly diverge from the overall trend; such points perhaps should be viewed as outliers and not cause a segment to be split further. Thus we employ an outlier detection test based on residuals[89]. To detect the presence of outliers, we assume that the trend can be represented as a regression line; thus all the points within the segment can be represented as $y_i = b + a\,x_i + e_i$ where $a$ and $b$ are calculated from least squared regression. The residual is $e_i = y_i - b - a\,x_i$ and the estimated standard deviation of $e_i$ is

$$s_i = \hat{\sigma}\sqrt{1 - \frac{1}{n} - \frac{(x_i - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

where $\hat{\sigma} = \sqrt{\sum e_i^2/(n-2)}$. If $\hat{\sigma}$ equals $0$, there are no outliers; otherwise, the standardized residuals $r_i = e_i/s_i$ are computed and $R_m = max|e_i/s_i|$ is used as a test statistic for outlier detection. We use a significance level of $\alpha = 0.01$ and the critical value given in [89]. If $R_m$ is greater than the critical value, outlier detection suggests the presence of an outlier in the sampled data points. If there are multiple $r_i$ that exceed the critical value, then multiple outliers are suggested. We use two features – the result of the outlier test and the number of outliers detected – in our feature base.

#### 6.2.4.5 Other local features

Besides the statistical tests and their corresponding results, other features which help describe the characteristics of the segment are also recorded and passed to the classifier to make a decision. They include:

- Number of data points in the current segment. Hypothesis tests such as the Q-test are sensitive to the number of data points. So this feature is used by the classifier to incorporate the consideration of sample size.

- We hypothesize that the variance in the segment may influence human perception on the split/no-split decision, so we consider the standard deviation of residuals in the segment rescaled by the horizontal length of the segment. If we represent the residuals as a column vector $\mathbf{r}$, this local feature is simply calculated as:

$$r_n = \frac{\sqrt{\mathbf{r}^T\mathbf{r}/n}}{(x_j - x_i)}$$

  where $n$ is the number of data points in the segment, $x_i$ and $x_j$ are the $X$ coordinates of the two end points of the segment, and $x_j > x_i$. The reason for rescaling the standard deviation by the horizontal length is that we are dealing with line graphs of different sizes, and we hypothesize that the standard deviation of residuals per unit length may affect visual perception.

- Standard deviation of the perpendicular distance between the data points and the regression line for the segment, rescaled by the length of the regression line between the two end points of the segment, which is $r_n \cos^2 \alpha$, where $r_n$ is defined as above and $\alpha$ is the angle between the regression line and the x axis (so $\tan \alpha$ is the slope of the regression line). This feature captures the rescaled deviation existing in the segment from a perpendicular perspective.

### 6.2.5 Global features

As opposed to other segmentation algorithms which only consider local information obtained solely from the segment under consideration, we also include global features that enable the classifier to consider the individual segment within a larger environment. Global features capture information about the whole graph rather than just the segment being examined. For example, the relative length of the segment compared with the whole graph can be used as a global feature. We hypothesize that global features may play an important role in deciding whether to split a segment. Thus the following attributes are included:

- The total number of data points in the whole line graph.
- The relative length of the current segment as a percentage of the whole graph. This feature is included to capture a global view of the segment with respect to the whole graph.

### 6.2.6 Support Vector Machine as classifier

To produce a training set, each graph in our corpus must be collected from popular media, scanned and sampled, and the ideal segmentation identified by human viewers. These are very time-consuming tasks. Therefore the size of the training set is limited. There are 18 features associated with each training instance, so the feature space is an 18 dimensional space.

Generally we can choose any classifier appropriate for the problem and the feature base used. For our segmentation problem, we chose a support vector machine as classifier because it works very well with high-dimensional data and a relatively small training set and avoids the curse of dimensionality problem[85]. SVM also lessens the chance of overfitting by using the maximum margin separating hyperplane which minimizes the worst-case generalization errors[5, 85]. Furthermore, as opposed to local methods such as nearest neighbor which require locating a small neighborhood for each new test instance,

the SVM can build the global hyperplane once from the training set and apply it to test cases with little computation.

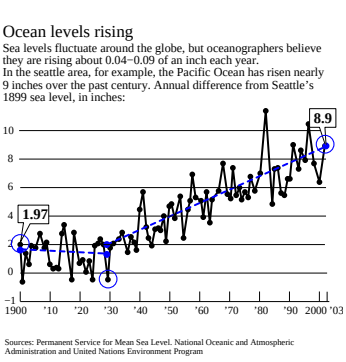
The support vector machine provides a maximum margin hyperplane to divide the 18 dimensional space into two parts. In our project, the feature vector is first normalized, and the linear kernel is applied to the feature space to generate a linear hyperplane. We use a linear kernel because an inappropriately chosen degree of polynomial kernel which generates a nonlinear hyperplane might induce overfitting.

We collected a corpus of 234 line graphs and built our training set from this corpus. Each line graph was entered as one instance in the training set along with the appropriate split or no-split decision. In the case of a split decision, each resulting segment is entered as an instance in the training corpus, along with their respective split or no-split decisions, and the process is recursively repeated. This produced a corpus of 649 segments; the feature values were recorded for these segments which were then used to train our decision module.

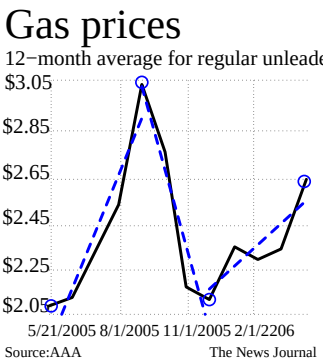
For example, consider a line graph which is eventually split into three final segments. Initially, the whole line graph is recorded as one training instance; after the first segmentation, two subsegments are generated and recorded as two new training instances; one of these new segments is again split, producing two more subsegments as new training instances. The segmentation process ends up with a total of five segments entered into the training set – two intermediate segments with split decisions and three segments with no-split decisions.

### 6.3 Examples

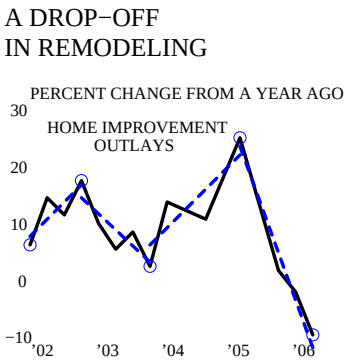
Figure 6.11 displays three examples of segmentations produced by our graph segmentation system. The three line graphs come from three different sources, *USA Today*, *BusinessWeek*, and a local newspaper, and differ from one another with respect to the number of trends and the amount of variance in each trend. The original line graphs




(a) A line graph conveying two trends with high variance




(b) A line graph conveying three trends with low variance




(c) A line graph conveying four trends with moderate variance

Figure 6.11: Three examples of segmentations produced by our graph segmentation system. The solid lines are the original line graphs, the small circles are the split points, and the dashed lines are the regression lines for the resulting trend segments.

are plotted with solid black lines. The split points identified by our segmentation algorithm are shown as circles (the start and end point of the line graph are also shown as circles because they are the boundaries of the beginning and ending segments). The high-level trends are located between each adjacent pair of splitting points, represented by dashed regression lines. We can see from the results that our segmentation algorithm accurately segmented the line graphs with different variances into visually apparent high-level trends.

## 6.4 Evaluation

The line graphs we collected for the graph segmentation evaluation came from multiple sources including *USA Today*, *Businessweek*, *Newsweek*, *the Wilmington News Journal*, etc. Table 6.1 lists the distribution of the number of segments of the line graphs from different sources. Because the three main sources are the *USA Today*, *Businessweek* and *Newsweek*, we have named all the other sources as "Other Sources".

To evaluate our graph segmentation methodology, we performed three evaluation experiments: cross-validation of the learned decision module which is responsible for making a split/no-split decision, and two human subjects experiments which evaluated

| Sources | 1 | 2 | 3 | $\geqslant$4 |
|---|---|---|---|---|
| *USA Today* | 49.0% | 35.7% | 12.2% | 3.1% |
| *Businessweek* | 29.7% | 36.3% | 20.9% | 13.2% |
| *Newsweek* | 29.4% | 47.1% | 17.6% | 5.9% |
| Other Sources | 32.0% | 32.0% | 28.0% | 8.0% |

Table 6.1: Distribution of the number of segments of the line graphs from different sources

the entire segmentation algorithm incorporating both the splitting module and the decision module.

### 6.4.1 Evaluation of the decision module

Since our corpus is small, we use leave-one-out cross validation to test the accuracy of the decision module, where each instance is used once as a test case and all the other 648 instances are used as training cases. The results of all 649 experiments are averaged together to obtain the accuracy of the model.

The accuracy obtained from leave-one-out cross validation is 88.3%, compared with the 67.2% accuracy of the baseline decision of no-split (the decision for the majority of instances in the training set). Thus our algorithm has a 31.4% improvement over the baseline. Table 6.2 gives the confusion matrix; it shows that our model is slightly biased toward avoiding splits when one should be made.

| | classified as no-split | classified as split |
|---|---|---|
| actual no-split | 410 | 26 |
| actual split | 42 | 171 |

Table 6.2: Confusion matrix

To identify the importance of the 18 features used by our support vector machine, we applied the recursive feature elimination(RFE) algorithm introduced by Guyon et. al.[40]. It measures the discriminating ability of the attributes by comparing their weights for the corresponding dimension of the hyperplane. It first calculates the weight vector of the SVM which produces the hyperplane that maximizes the margin, and recursively

1) eliminates the feature with the lowest absolute weight in the representation of the hyperplane and 2) rebuilds the hyperplane, until all attributes have been removed one by one. The earlier an attribute is removed, the less discriminating it is and thus the lower its rank. Table 6.3 lists the features in rank order from most significant to least significant.

| Rank | Feature name |
|---|---|
| 1 | rescaled standard deviation of perpendicular distance |
| 2 | relative length of current segment |
| 3 | difference between actual runs and mean runs |
| 4 | rescaled standard deviation of vertical distance |
| 5 | correlation coefficient |
| 6 | number of points in current segment |
| 7 | Q-test in 995 significance level |
| 8 | Q-test in 95 significance level |
| 9 | runs test |
| 10 | standard deviation of runs |
| 11 | outlier detection |
| 12 | Q-test in 99 significance level |
| 13 | F-test in 95 significance level |
| 14 | mean runs |
| 15 | total number of points |
| 16 | number of outliers |
| 17 | actual runs |
| 18 | F-test in 90 significance level |

Table 6.3: Features listed in rank order

Let us examine the top ranked features. The first and fourth features measure the standard deviation of the segment in unit length, either from a perpendicular or a vertical perspective. The rank of these two features indicates that the rescaled standard deviations within a segment play a very important role in making a split decision. The second feature captures global information by measuring the relative length of the current segment; it reveals the fact that when a split/no-split decision is made, features based on local information in the segment are not enough. For identifying trends that are visually apparent to humans, we must consider the segment in a larger context. Although we only have one global feature among the top ten features, it plays an important role in the model,

and this global information is ignored by other time series segmentation algorithms. In future work, we will consider other global features, such as the relative location of the segment. In addition to the two standard deviations and the relative length of the segment, the correlation coefficient, Q-test, Runs Test, and number of points in the segment all rank among the top ten features and thus play a significant role in the model.

It is interesting to note that the features coming from the F-test and outlier detection are not ranked in the top ten. This indicates that although the F-test is also a change point detection statistic, it is not as powerful as the Q-test for our segmentation task.

### 6.4.2 Evaluation of the entire segmentation algorithm

Recall that our leave-one-out cross validation tested our decision module's ability to make split/no-split decisions on individual segments. To test how well our segmentation algorithm segments entire line graphs into visually apparent trends (both deciding when to split segments and where to split them), we used seven human evaluators. The human subject evaluation had two parts. The first part, described in Section 6.4.2.1, compared the segmentations produced by our system with those produced by another segmentation method from the literature. The second part, described in Section 6.4.2.2, was a qualitative evaluation of our graph segmentations. In both experiments, our segmentation for each graph was produced by a model constructed from the other 233 graphs, thus avoiding the problem of biasing the results by including the test graphic in the corpus used to train the model.

#### 6.4.2.1 Comparative experiment

In this comparative experiment, seven human evaluators were each given 234 line graphs with two segmentations, one produced by our segmentation algorithm and the other produced by the comparative segmentation algorithm.

The comparative algorithm reflects existing approaches to time-series segmentation based on error minimization[41, 46, 39, 46] and looks for a predefined $k$ number of

segments to minimize the residual sum of squares of the piecewise linear regression. This error reduction algorithm can be implemented top-down or bottom-up. To be comparable with our top-down segmentation approach, we used the top-down one.

A critical aspect of this method is how to define the number of segments $k$ since the same $k$ for all line graphs is not appropriate. Salvador et. al.[80] suggest a method for identifying the appropriate $k$ for a given line graph by locating the *knee* of the plot of the residual sum of squares against the number of segments, as in Figure 6.12 which is called an evaluation graph.

The *knee* is the point which generates a two-phase regression in the evaluation graph and minimizes the residual sum of squares of the evaluation graph. The evaluation graph is generated by a top-down iterative process in which one splitting point (i.e. one segment) is added in each iteration. On each iteration, a new splitting point is added such that it results in the largest reduction of $R(Y, \hat{Y})$, thereby generating one more segment. $R(Y, \hat{Y})$ is the risk(expected loss) between real data $Y$ and the estimated data $\hat{Y}$ using the least squares regression with the newly added splitting point. After the $n-2$ intermediate points are all added as splitting points where $n$ is the total number of points in the line graph, there will be $k = n - 1$ segments. An evaluation graph, such as the one shown in Figure 6.12, plots $R(Y, \hat{Y})$ for each number of segments. Then the optimal number of segments is determined by the location of the *knee* in the evaluation graph of errors versus number of segments, where the *knee* is a change point in the evaluation graph, identified by fitting a two-phase regression model which minimizes the residual sum of squares in the evaluation graph. The number of segments corresponding to the identified *knee* is used as the number of segments $k$ for the comparative algorithm.

For each line graph, the human evaluator was given two segmentations, one produced by our segmentation algorithm and one produced by the above comparative algorithm. The order of the appearance of the two segmentations was randomly assigned, to avoid bias resulting from the order of presentation. The evaluators had three options:

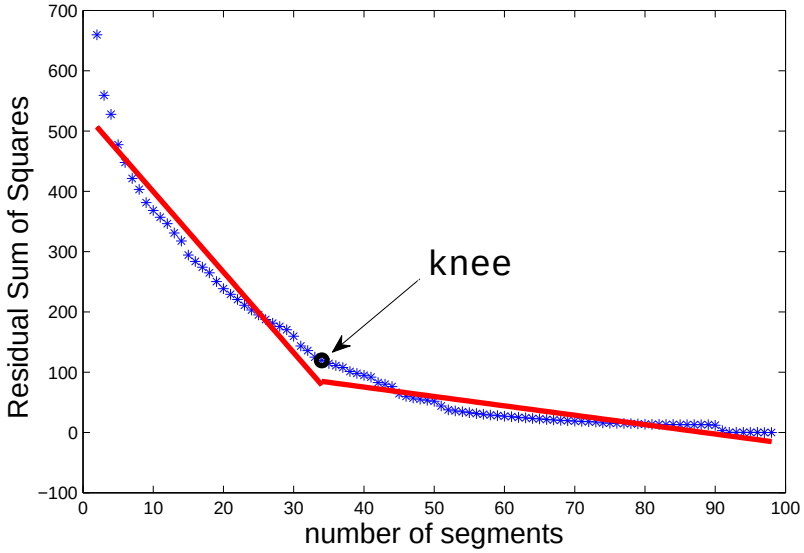



Figure 6.12: A plot of segmentation errors against number of segments

"The segmentation on the left is better", "The segmentation on the right is better", and "I have no preference".

For 218 of the 234 line graphs, a majority of the evaluators had the same response; on only 16 line graphs was there no majority decision, indicating that these line graphs had differences that made it difficult to identify the better segmentation. For the 218 line graphs where there was a majority decision, the segmentation produced by our system was preferred for 76.1% of the graphs, there was no preference for 8.8% of the graphs, and the segmentation produced by the comparative algorithm was preferred for 15.1% of the graphs. Thus we see that the segmentations produced by our system were preferred five times more often than the segmentations produced by the comparative algorithm.

We also computed how often each evaluator preferred our segmentation, had no preference, or preferred the segmentation produced by the comparative algorithm. The preference data for each individual evaluators are shown in Table 6.4, with the last two rows showing the mean and standard deviation of the percentage evaluation. All evaluators preferred our segmentation more often than they preferred the other segmentation.

| Evaluator | Our Algorithm | Comparative Algorithm | No Preference |
|---|---|---|---|
| 1 | 72.5% | 20.6% | 6.9% |
| 2 | 60.1% | 36.9% | 3.0% |
| 3 | 85.4% | 5.6% | 9.0% |
| 4 | 56.7% | 17.6% | 25.8% |
| 5 | 59.2% | 23.2% | 17.6% |
| 6 | 50.6% | 31.8% | 17.6% |
| 7 | 82.8% | 11.6% | 5.6% |
| Mean | 66.8% | 21.0% | 12.2% |
| SD | 13.6% | 10.9% | 8.3% |

Table 6.4: Preference table against comparative algorithm

Averaging the results for the seven evaluators, we find that our system performed better than or equal to the comparative algorithm 79% of the time.

These results show that our learned graph segmentation algorithm produces better segmentations of line graphs into visually apparent trends than does a traditional algorithm based on error minimization.

#### 6.4.2.2 Qualitative evaluation of segmentations

Seven human evaluators were given 254 line graphs along with candidate segmentations; 234 of the line graphs were the ones in our corpus with the segmentations produced by our system and 20 were additional line graphs with bad segmentations. The latter were scattered throughout the evaluation set and were included to avoid bias by the evaluators. The evaluators were not told that intentionally bad segmentation examples had been included in the evaluation set. The evaluators were asked to assign a score between 1 and 5 to each segmentation as shown in Table 6.5.

The average rating for the segmentations produced by our system was $4.25$ with $0.55$ standard deviation across the 234 line graphs, showing the performance of our segmentation algorithm is between "Very good" and "Ideal". The 20 extra graphs with bad segmentations received an average rating of $1.57 \pm 0.44$ which is between "Terrible" and "Poor". These results verify that our graph segmentation algorithm successfully segments

| Rate | Description |
|---|---|
| 5 | Ideal |
| 4 | Very Good |
| 3 | Acceptable |
| 2 | Poor |
| 1 | Terrible |

Table 6.5: 5-points rate of the segmentation

line graphs into visually apparent trends. This good performance is a result of the learned model generated by our machine learning framework.

## 6.5 Summary

This chaplter has presented the Graph Segmentation Module, whose goal is to segment the line graph into a series of visually distinguishable trends. Our methods provides significantly better results than an error reduction based method with heuristics determining the number of segments. In particular, our method uses a set of features, including both local and global features, with a SVM classifier to provide a learned model rather than using error minimization with a pre-determined threshold chosen by a prior domain knowledge.

The segments produced by the Graph Segmentation Module correlate with the visual descriptions proposed by [82, 71]. They are thus used as the discrete entities which are used to propose candidate meassages fro consideration by the Bayesian network, as discussed in Chapter 7.

# Chapter 7

# RECOGNIZING THE INTENDED MESSAGE OF LINE GRAPHS WITH A BAYESIAN NETWORK

In Chapter 5, we argued that a Bayesian network would be an appropriate inference mechanism for identifying the intended message of a line graph. This chapter presents the Bayesian network. Section 7.1 discusses how candidate messages are constructed. Section 7.2 discusses the structure of our Bayesian network. Section 7.3 presents several examples of graphs that are processed by our system and Section 7.4 presents an evaluation of the system.

## 7.1 Generating intended message candidates

Given that our inference mechanism will be a Bayesian network, we need to construct a set of candidate intended messages that the Bayesian network will choose among. Each candidate message will consist of one of the ten high-level message categories identified in Chapter 3 along with instantiations of each of its parameters. As argued in Chapter 5, it is unreasonable to use every sample point (or even every short segment of a very jagged line graph) as a possible instantiation of these parameters. Thus the Graph Segmentation Module described in Chapter 6 was designed to segment the line graph into a sequence of visually distinguishable trends. The boundary points of these trend segments will be used as parameters for the intended message candidates. The module that produces the candidate messages is called the Suggestion Generation Module.

Before generating the appropriate suggestions, the Suggestion Generation Module needs to do some preprocessing:

1. It needs to categorize every segment detected by the Graph Segmentation Module as a "rise", "fall" or "stable" trend. The reason for doing this is so that only appropriate segments are used to generate each kind of intended message. For example, we would not want to instantiate the parameters of a Rising-Trend message category with the end points of a segment that is actually falling.

   Rising segments can be differentiated from falling segments by the slope direction of the least square linear regression across the segment. Difficulty arises in determining whether to classify a segment as stable. Instead of assigning a hard threshold to the slope of the regression line, we measure the 95% confidence interval of the regression line. If the upper bound has a slope where the sign is different from that of the lower bound, the segment is classified as "stable"; otherwise, it is classified as "rise" or "fall" according to the sign of its slope. This method is much better than using a hard threshold. For example, if we use a hard threshold such as $\pm 5°$, then the line graph in Figure 7.1 will be categorized as a stable trend segment since the slope of its regression line is $-4.04°$. But the upper bound and lower bound of the 95% confidence interval of the regression line both have a negative slope and thus it is classified as a falling segment with very small slope.

2. We hypothesize that small changes at the end of a line graph, as in Figure 7.2, may be particularly salient to a viewer, especially if they represent the value of an entity near the current time. However, the Graph Segmentation Module will most likely smooth such small changes into an overall longer smoothed trend. Thus, a short routine based on the F-test examines the end of the line graph and if it represents a change in slope from the preceding points, that short portion is treated as a separate segment. This short segment (if any) is added to the result produced by the Graph Segmentation Module so that it can be used in constructing Change-Trend-Last-Segment candidate messages.

   The suggestion generation process proceeds as follows:

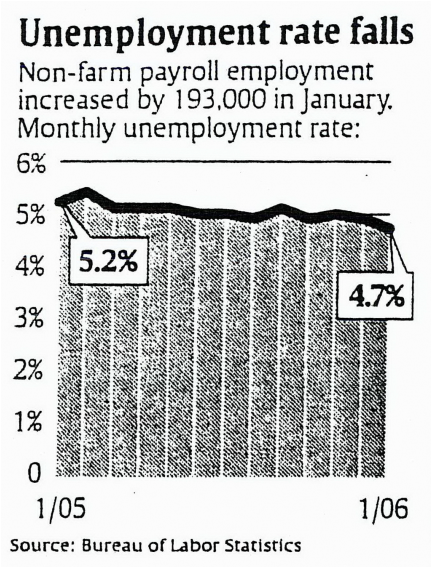



Figure 7.1: Line graph with a falling trend, but the slope is only $-4.04°$

- Generate a Rising-Trend or Falling-Trend candidate message for each rising or falling segment respectively. The parameters are instantiated with the beginning and ending points of this single segment.

- Generate one Change-Trend suggestion for each pair of adjacent segments belonging to different trend types. The parameters are the beginning point of the first segment, the change point between segments, and the ending point of the last segment.

- Generate one Change-Trend-Return suggestion for each series of three adjacent segments. The parameters are the beginning point of each of the three segments and the ending point of the last segment.

- Generate one Change-Trend-Last-Segment suggestion for the pair of adjacent segments if the second segment touches the end of the line graph. The parameters are

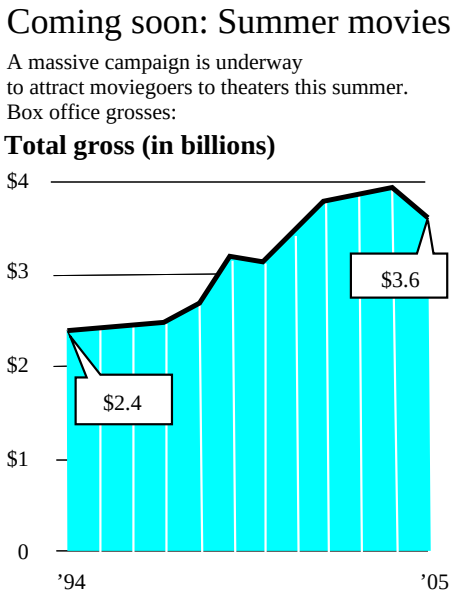



Figure 7.2: Line graph from a local newspaper

the beginning point of the first segment, the change point between segments, and the ending point of the last segment.

- Generate one Contrast-Segment-Change-Trend suggestion for the series of three adjacent segments if the third segment touches the end of the line graph. The parameters are the beginning point of the each of the three segments and the ending point of the last segment.

- Generate one Big-Jump or Big-Fall suggestion for each rising segment or falling segment respectively if the segment is not the last segment of the graph. (It is because we assume that a Big-Jump has a Big-Jump-Sustain or a Big-Jump-NotSustain as a sub-intended message which requires a segment following the Big-Jump segment; the same assumption also applies to Big-Fall. This will be discussed further in Section 7.5) The parameters are the beginning point of the sudden rising or falling segment, the ending point of this segment, and the ending point of the line graph.

- Generate one Point-Correlation intended message for the whole line graph.

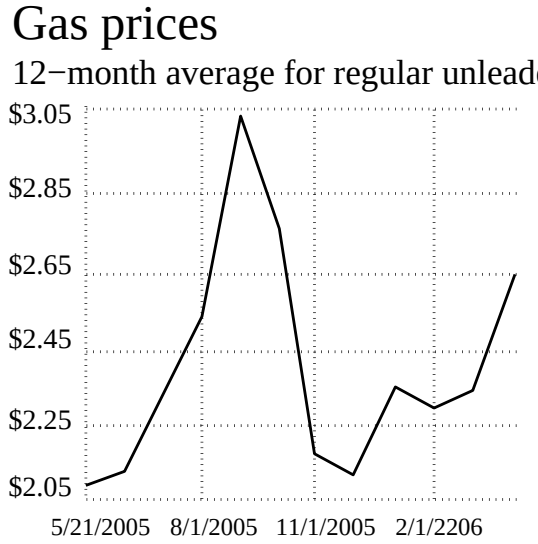



Figure 7.3: Line graph from a local newspaper

.

Consider, for example, the graphic in Figure 7.3. The Graph Segmentation Module produces a sequence of three visually distinguishable trends. The Suggestion Generation Module proposes the 11 possible messages shown in Table 7.1[1]:

RT (5-21-05, rise, 9-1-05)
CT (5-21-05, rise, 9-1-05, fall, 12-1-05)
RT (12-1-05, rise, 4-25-06)
CT (9-1-05, fall, 12-1-05, rise, 4-25-06)
FT (9-1-05, fall, 12-1-05)
CTR (5-21-05, rise, 9-1-05, fall, 12-1-05, rise, 4-25-06)
BJ (5-21-05, rise, 9-1-05)
CTLS (9-1-05, fall, 12-1-05, rise, 4-25-06)
BF (9-1-05, fall, 12-1-05)
CSCT (5-21-05, rise, 9-1-05, fall, 12-1-05, rise, 4-25-06)
PC (5-21-05, rise, 9-1-05, fall, 12-1-05, rise, 4-25-06)

Table 7.1: The 11 possible messages generated for Figure 7.3

[1]Our system works with the actual points in the graph; for clarity of presentation, we only show the x-values for the points corresponding to $<\text{param}_i>$ in Table 3.1.

## 7.2 Bayesian network inference

A Bayesian network[70, 48] is a probabilistic graphical model for inferring a conclusion from observations. It uses knowledge of the likelihood $p(\text{child nodes}|\text{parent nodes})$ and the prior probability of each parent node calculated from the training set to compute the posterior probability of a parent node using Bayes rule:

$$p(\text{Parent}|\text{Child}) \propto p(\text{Child}|\text{Parent})p(\text{Parent}) \tag{7.1}$$

Our Bayesian network is a tree structure that is generated as a top down process:

1. Root Node: The top level node is the root node and contains each of the ten intended message categories as its possible values.

2. Category Nodes: The second level consists of one node for each intended message category in the root node at the top level. The ten message category nodes are generated as the children of the root node. They are just drawn out of the top level node for ease of representation so that the complexity of the conditional probability table for both the root node and the message category nodes can be reduced.

3. Candidate Message Nodes: Beneath each of the Category Nodes are a set of candidate message nodes, one for each instantiated candidate message in that message category. The candidate message nodes represent the suggestions generated by the Suggestion Generation Module. If there are any suggestions belonging to some intended message category, the candidate message nodes are attached as child nodes of the corresponding category node. Since only one suggestion belonging to an intended message category can be the true intended message, we want the multiple suggestions generated for the same intended message category to compete for all the probability calculated for the intention category node. To accomplish this, we add inhibitory links among the multiple suggestions generated for the same intended message category. Figure 7.4 shows part of the top three levels of the

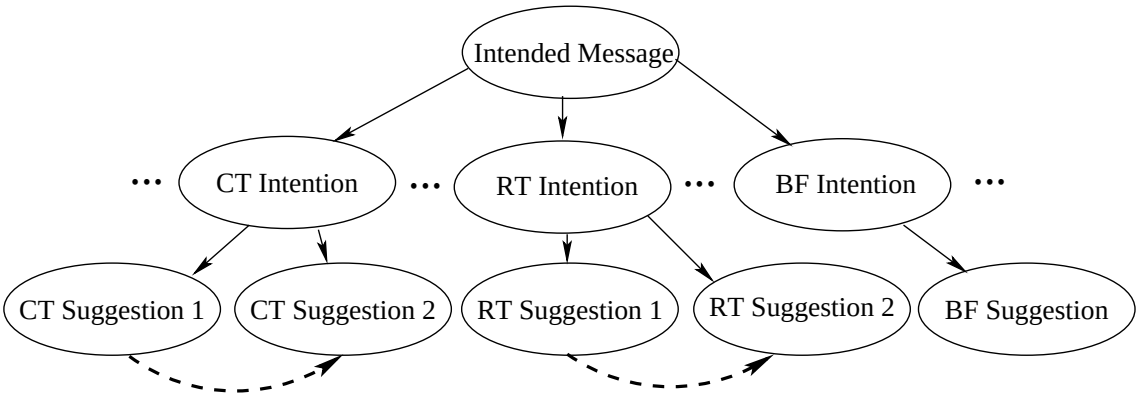



Figure 7.4: The top three levels of our Bayesian network

Bayesian network constructed for Figure 7.3, with the dashed lines representing the inhibitory links.

4. Evidence Nodes: Evidence nodes capture the evidence for or against a message category or a candidate intended message. As discussed in the next section, evidence specific to a candidate intended message is attached as a child of that node whereas evidence for or against a message category is attached as a child of the top level root node. Figure 7.5 shows multiple evidence nodes at different levels of the Bayesian network, plotted as dashed ovals.

When the evidence is entered and the Bayesian network updates its belief, the Top Level Node contains the posterior probability of the 10 intended message categories. The intended message category with the highest probability is selected, and the intended message within this category with the highest posterior probability is regarded as the final intended message for the line graph.

### 7.2.1 Extracting evidence and entering it into the Bayesian network

The evidence nodes in the Bayesian network record the communicative signals present in the line graph. Table 7.2 through Table 7.3 list the evidence nodes and their corresponding possible values for a candidate message node.

Table 7.2: The evidence nodes and their corresponding values

| Name of Node | Description and values |
|---|---|
| Percentage | Records the proportion of the entire line graph that is covered by the candidate message. It can take values: $100\%$, $\geq 80\%$, $< 80\%$. |
| EndpointsAnnotated | Records whether the two endpoints of a candidate message are annotated. It can take values: none,one,both. |
| SplittingAnnotated | Records whether the splitting points of a candidate message are annotated. It can take values: none, some, all. |
| OtherAnnotated | Records whether there are any other annotations not on either endpoints or splitting points of a candidate message. It can take values: none, one, more. |
| NumMiddleAnnotations | Records the number of annotations which appear on points other than the two ends of the line graph. It counts all the annotations on either the splitting points or other points, except the two endpoints. It can take values: none, one, more. |
| EndpointsSalient | Records whether the endpoints are salient. Salience means that the points are referred to by a word or words in the caption or description in the line graph. It can take values: none, one, both. |
| SplittingSalient | Records whether the splitting points are salient. It can take values: none, some, all. |
| OtherSalient | Records whether there is any other salient point other than endpoints and splitting points. It can take values: none, one, more. |
| LastLength | Records the ratio of the length of the last segment with the length of the whole line graph, measured on the horizontal axis. It can take values: $\geq 20\%$, $< 20\%$. |
| TouchEnd | Record whether this candidate message covers the right end of the line graph. It can take values: yes, no. |
| LastSegmentMatchWord | Records whether the direction of the last segment matches the word category in caption/description. For example, if a verb in the caption is classified into class 1 in Table 7.5 and the last segment is also a rising segment, then this feature takes the value "Match". It can take values: Match, NoMatch. |

Table 7.3: The evidence nodes and their corresponding values - continued

| Name of Node | Description and values |
|---|---|
| YScale | Big-Jump/Big-Fall only. Measures the ratio of the height of a segment with respect to the height of the whole line graph. It can take values: $\geq 80\%$, $< 80\%$. |
| XDuration | Big-Jump/Big-Fall only. Measures the ratio of the width of a segment with respect to the width of the whole line graph. It can take values: $\geq 25\%$, $< 25\%$. |
| BigSlope | Big-Jump/Big-Fall only. Measures the absolute value of the slope.. It can take values: $\geq 60°$, $< 60°$. |
| SmallLargeRatio | Big-Jump/Big-Fall only. Measurses the ratio between the length of the smallest segment and the length of the largest segment. The lengths are measured along the horizontal axis. It can take values: $\geq 0.2$, $< 0.2$. |

Table 7.4: The evidence nodes for the words in caption/description and their corresponding values

| Name of Node | Description and values |
|---|---|
| CaptionVerb | Record the category of the verbs in caption. It takes seven values for the seven classes in Table 7.5. |
| CaptionNoun | The category of the nouns in caption. |
| CaptionAdjective | The category of the adjectives in caption. |
| DescriptionVerb | The category of the verbs in description. |
| DescriptionNoun | The category of the nouns in description. |
| DescriptionAdjective | The category of the adjectives in description. |

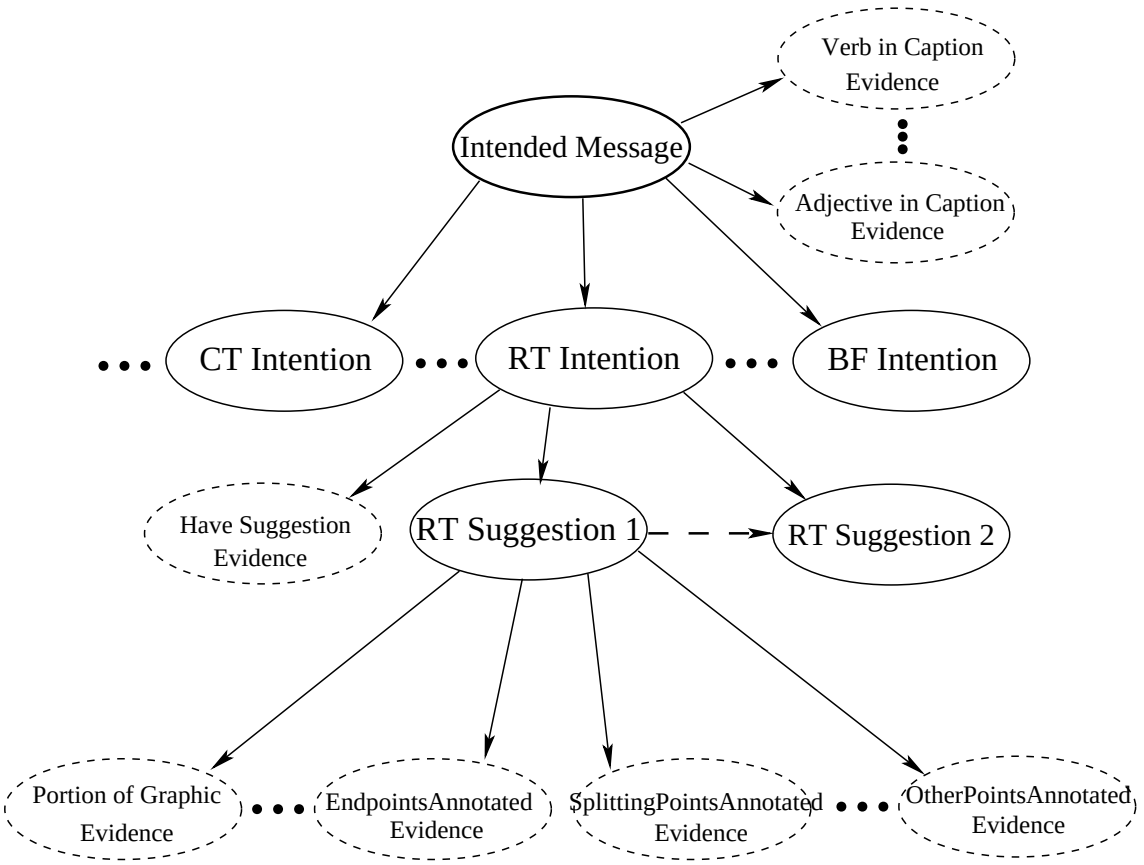



Figure 7.5: Bayesian Network with some evidence nodes

As we have discussed, the nouns, verbs and adjectives in the caption and description of a line graph serve as communicative signals that suggest an intended message category. Extending work by Seniz Demir and ideas espoused by Elzer[28], we employ a Caption Tagging Module which utilizes a part-of-speech tagger to process the caption and description of a line graph and extract potentially helpful nouns, verbs and adjectives. For example, the caption "ocean levels rising" contains the verb "rise", which might suggest a Rising-Trend or a Change-Trend message where the second segment is rising. Using WordNet, we identified potentially helpful verbs and organized them into classes of similar verbs. For example, the verbs *"jump"* and *"boom"* reside in one verb class, whereas the verbs *"resume"* and *"recover"* reside in a different verb class. Similar classes were constructed for nouns and adjectives/adverbs. In our research, we categorized the helpful

words into six categories, plus a category indicating that there is no helpful word. Table 7.5 shows the categories and several example words in each category. The presence or absence of words in the word categories is used as evidence for or against an intended message category. This evidence appears as children of the top level node in the Bayesian network.

| | |
|---|---|
| class 1 | *rise increase ascend grow boom* |
| class 2 | *fall descend decline plunge* |
| class 3 | *remain stay* |
| class 4 | *change rebound recover* |
| class 5 | *fluctuate swing* |
| class 6 | words from class 1 and class 2 both appear in the same caption or description |
| class 7 | The caption or description doesn't contain a helpful word |

Table 7.5: The word categories and several example words in each category.

Having a candidate suggestion for a particular category strengthens the posterior probability of that intended message category. Thus we add an evidence node named "HasSuggestion" as a child node of the category node. Whenever there are no suggestions generated for an intended message category, the "HasSuggestion" node will have finding "No" instantiated, which is evidence against that intended message category.

### 7.2.2 Constructing the conditional probability tables

Associated with each node in a Bayesian network is a conditional probability table that reflects the probability of each of the values of that node given the value of the parent node. (The probability table for the top-level node captures the prior probabilities of each of the intention categories.) The conditional probability tables for our Bayesian network are computed from an analysis of our corpus of 240 line graphs.

Three coders had previously identified the intended message of each line graph in our corpus. If there was disagreement among the coders on either the message category or the instantiation of the parameters, we used consensus-based annotation[1] in which the

coders discussed the graphic and made a consensus decision about the graphic's intended message. As noted in [1], this allows us to include the hard graphics whose intentions are less clear and about which the coders might initially have different opinions.

We constructed a spreadsheet capturing all of the information needed to construct the conditional probability tables from our corpus. For each graphic, all of the candidate messages generated by the Suggestion Generation Module were entered into the spreadsheet. If one of these messages matched the intended message of the graphic as identified by the coders, it was marked as Intended-Message. Otherwise, the segmentation was done manually and the suggestions generated from this segmentation were inserted into the spreadsheet, with the correct suggestion marked as Intended-Message for the graphic.[2] For each of these messages, the values for the evidence listed in Table 7.2, 7.3 and 7.4 plus the "HasSuggestion" evidence were recorded.

Formulas were constructed for computing all of the required conditional probability tables from the information in the spreadsheet. To overcome the impact of probability values that are 0 according to the corpus, we applied smoothing and replaced them with a value of $0.01\%$ and then normalized each probability table to make the sum still be 1.

One such conditional probability table is shown in Table 7.6. It gives the conditional probability that the endpoints $<\text{param}_1>$ and $<\text{param}_2>$ of a Rising-Trend($<\text{param}_1>$, $<\text{param}_2>$) message are annotated in the graphic, given that the intended message is (or is not) a Rising-Trend. For example, the InPlan column of the conditional probability table shows that the probability that both endpoints are annotated is 55.4% if a Rising-Trend is the intended message, and the NotInPlan column shows that the probability is 3.6% if it is not the intended message.

[2]Note that when the intended message is not one of the messages produced by the Suggestion Generation Module, our system cannot succeed on that graphic since the correct message will not be one of the messages considered by the Bayesian network. The purpose of manually segmenting the graphic and entering the correct message into the spreadsheet is solely to facilitate training the Bayesian network.

Endpoints Annotated Table

| Rising-Trend | InPlan | NotInPlan |
|---|---|---|
| Only one endpoint is annotated | 12.3% | 26.2% |
| Both endpoints are annotated | 55.4% | 3.6% |
| No endpoint is annotated | 32.3% | 70.2% |

Table 7.6: A sample conditional probability table

The conditional probability tables for the evidence in Table 7.4 were calculated more straightforwardly because these evidence nodes are attached to the Root Node. Given an intended message category, the conditional probability table for the possible values of this evidence, as shown in Table 7.5, is calculated as the smoothed distribution of the values in the training instances which belong to this intended message category. For example, within the training instances which have a Rising-Trend intended message, $32.1\%$ of them have a helpful verb in the caption which belongs to the class 1 word category, and $67.8\%$ of them have no helpful verb in the caption.

### 7.2.3 Processing a new graphic

After a line graph is processed by the Visual Extraction Module, the Graph Segmentation Module, and the Suggestion Generation Module as in Figure 5.1, the Bayesian network is then dynamically built for this graphic using Netica[69].[3] Netica then computes the posterior probability of each of the higher-level nodes based on the findings recorded in the leaf evidence nodes. The entry in the top-level node with the highest posterior probability represents the system's hypothesis about the category of intended message. The candidate message node that is a child of this intention category node with the highest probability is then selected as the system's hypothesis about the graphic's intended message.

[3]Notice that the structure of the Bayesian inference network for each line graph might not be the same.

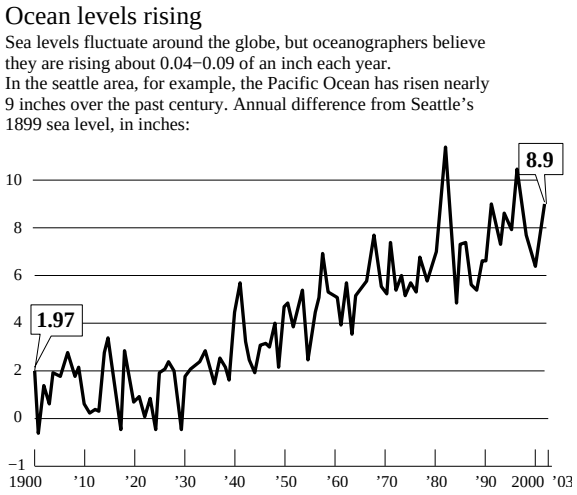



Figure 7.6: A line graph which our system draws a Change-Trend conclusion with 99.9% confidence, stable from 1900 to 1928 and rise from 1928 to 2003

## 7.3 Examples

In hypothesizing the graphic's intended message, our message recognition system takes into account the presence or absence of communicative signals that might appear in the line graph. The following examples illustrate the impact that these signals have on our system's hypothesis about the graphic's intended message.

The first example is shown in Figure 7.6. In this line graph, the caption has the verb "rise" as a communicative signal which is in class 1 in Table 7.5. In the graphic part, the two endpoints of this line graph are annotated. The Change-Trend candidate message suggested by Suggestion Generation Module covers the full length of the line graph. After the evidence nodes are instantiated and the belief is updated, the Bayesian network assigns the Change-Trend(1900,stable,1928,rise,2003) suggestion $99.9\%$ confidence and it correctly matches the intended message identified by the three human annotators.

Another of our examples is based on the line graph shown in Figure 7.7 which appeared in an article in a local newspaper. It was a difficult graph for our coders who were unsure whether it was intended to convey a Change-Trend in Durango sales over the period from 1997 to 2006 or whether it was intended to convey a Falling-Trend in

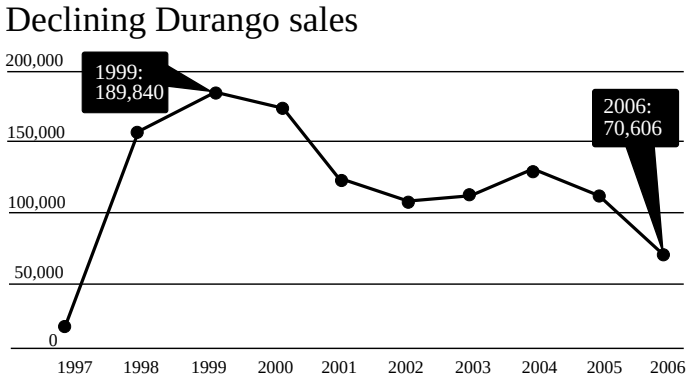



Figure 7.7: A line graph which appeared in an article in a local newspaper uses communicative signals from both caption and annotations which causes ambiguity for human annotators

Durango sales from 1999 to 2006. The coders eventually decided on the Change-Trend message.

Now let us examine the system's processing of the graphic. As discussed in Section 7.4, we use leave-one-out cross validation; thus the conditional probability tables in the Bayesian network used to process the graphic in Figure 7.7 are constructed from the data in the other 239 graphs in our corpus and do not include any information from the graphic in Figure 7.7.

The Graph Segmentation Module produces a sequence of two segments: a rising segment from 1997 to 1999 and a falling segment from 1999 to 2006. The Suggestion Generation Module generates several messages for consideration; the two of interest to us in this example are a Change-Trend(1997,rise,1999,fall,2006) and a Falling-Trend(1999,2006). There are a number of communicative signals in the graphic that were deliberately entered by the graph designer: 1) the annotation giving the value for the year 1999, 2) the annotation giving the value for the year 2006, and 3) the verb *"declining"*[4] in the caption *"Declining Durango sales"*. Other evidence entered into the Bayesian

[4]Although one might think *"declining"* as an adjective, the Part-of-Speech tagger used by our Caption Tagging Module tags it as a verb.

network includes (among others) the portion of the graphic covered by each candidate message, and the relative width of the last segment of each candidate message. For the Change-Trend message, the message covers the whole graphic and the last segment covers more than half of the graphic; for the Falling-Trend message, the last (and only) segment covers much, but not all, of the graphic.

The system considers all of the candidate messages and the evidence entered into the Bayesian network; it hypothesizes that the graphic's intended message is that there is a Change-Trend (rising from 1997 to 1999 and then falling from 1999 to 2006) in Durango sales and assigns this hypothesis a probability of 78.3%. The hypothesis that the graphic is intended to convey a Falling-Trend is assigned a probability of 20.4%. All the other candidate messages share the remaining 1.3% probability. The probabilities assigned to the Change-Trend and Falling-Trend messages reflect the ambiguity about the intended message that is inherent in the graphic. The presence of the verb *"declining"* and the annotations on both points that are parameters of the Falling-Trend message all seem to support the Falling-Trend intended message. However the graphic designer has a reason for including all of the data points in the graphic, and the fact that the Change-Trend message covers the whole line graph is evidence for it being the intended message of the graphic. Besides covering the whole line graph, two of the three points that are parameters of the Change-Trend message are annotated. These pieces of evidence caused the system to prefer the Change-Trend message over the Falling-Trend message, although with a large amount of uncertainty as reflected by the probability assigned to the Falling-Trend message.

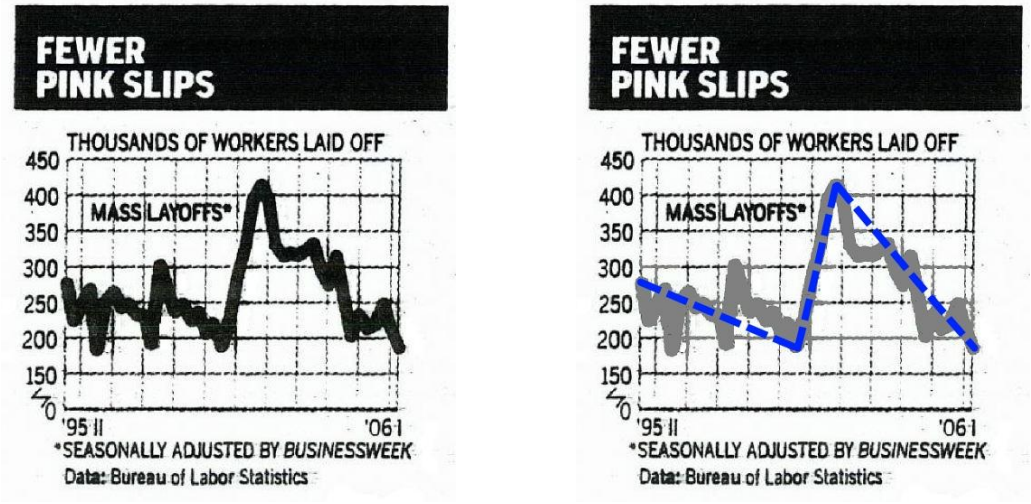
Now let us examine how the system's hypothesis changes as we vary the communicative signals in the graphic. First, let's change the caption to just *" Durango sales"*. The system now hypothesizes that the graphic is intended to convey the Change-Trend message and assigns this hypothesis a probability of 92.4%; the Falling-Trend hypothesis is assigned a probability of only 7.0%. Note that there is still some confusion about the

intended message due to the two points that are annotated in the graph. Now suppose that we change the caption to *"Durango sales change"*. Whereas the verb *"declining"* might be used in the caption of a Change-Trend message, it is less likely that the verb *"change"* would be used with a Falling-Trend message. The system's hypothesis doesn't change, but the probability assigned to the changing trend message increases from 92.4% to 97.5%. Here we see that the verbs/adjective evidence overcomes the confusion caused by annotating only two points in the graphic.

Now let's return to the original graphic in Figure 7.7 and analyse how the intended message changes according to annotation communicative signals. We still use Figure 7.7 with the caption as "Declining Durango Sales", but suppose that we add an additional annotation giving the value of Durango sales in 1997. The three annotations match exactly the three parameters of a Change-Trend message. Now the system's hypothesis changes dramatically — it identifies the Change-Trend candidate as the intended message of the graphic and assigns it a probability of 99.8%, with the Falling-Trend message assigned a probability of 0.2%. Note that although the verb *"declining"* is most associated with a Falling-Trend message, it can also be used with a Change-Trend message to draw attention to the falling portion of the changing trend.

These examples show the advantage of using a probabilistic inference model. Instead of classifying a line graph into a hard category, the Bayesian network can assign a probability to each candidate intended message, which gives us the capability of measuring the ambiguity of the line graph's intended message and providing a secondary candidate which might also be reasonable.

The third example illustrates more about recognizing the Big-Jump and Big-Fall intended messages. Consider Figure 7.8a; the annotators assigned this line graph a Big-Jump intended message, indicating that this line graph conveys a message that there was a big jump in the number of lay-off from the second quarter of 2000 to the third quarter of 2001. The Graph Segmentation Module generates a series of three segments as shown by




(a) (b)

Figure 7.8: An example line graph that was assigned a Big-Jump intended message by annotators

the dashed line in Figure 7.8b, and the Suggestion Generation Module generates 10 suggestions from the three segments. Among the 10 intended message candidates, there are two candidates that receive higher probabilities than the others. They are Big-Jump(2000 Quarter 2, 2001 Quarter 3) and Falling-Trend(2001 Quarter 3, 2006 Quarter 1). Let's take a look at the communicative signals for each of them. The caption of the line graph has a helpful adjective "fewer" and a supposedly helpful verb "slip"[5]; both belong to the class 2 word category in Table 7.5. In the Big-Jump candidate message, the second segment which is a sharp rising trend takes more than 80% of the height of the line graph and less than 25% of the width of the line graph. The Falling-Trend candidate message takes less than 80% of the width of the line graph and touches the end of it.

Our system assigns the Big-Jump candidate message a probability of 81.4%. Although our system correctly identifies the intended message of this line graph, it still

[5]Although "slips" should have been tagged as a noun, the part-of-speech tagger identified it as a verb.

assigns 12.1% probability to the Falling-Trend candidate message because the helpful adjective and verb in the caption both implicitly refer to a falling segment and are against a Big-Jump intended message. If we remove both words from the caption, the system's confidence changes. The Big-Jump(2000 Quarter 2, 2001 Quarter 3) candidate message now has 91.3% probability.

### 7.4 Evaluation of the system

We evaluated the performance of our system for recognizing a line graph's intended message on our corpus of 240 simple line graphs that were collected from various magazines such as *USA Today*, *Businessweek*, *the Wall Street Journal*, *the New York Times*, and from local national newspapers. Table 7.7 shows the distribution of different intended message categories in different sources. Because *USA Today*, *Businessweek*, *Newsweek* and *the Wall Street Journal* are the four major sources, we put all other sources into "Other Sources" category; they include *the New York Times*, *the Wilmington News Journal*, etc. Please notice that in Table 7.7 the sum within a column is equal to 100%.

| Messages | *USA Today* | *Businessweek* | *Newsweek* | *WSJ* | "Other Source" |
|---|---|---|---|---|---|
| RT | 38.0% | 21.7% | 29.4% | 0 | 23.8% |
| FT | 12.0% | 9.6% | 5.9% | 0 | 4.8% |
| ST | 1.0% | 0 | 0 | 0 | 0 |
| CT | 23.0% | 32.5% | 23.5% | 0 | 19.0% |
| CTR | 7.0% | 14.5% | 5.9% | 0 | 9.5% |
| CTLS | 7.0% | 8.4% | 17.6% | 0 | 0 |
| CSCT | 4.0% | 1.2% | 0 | 0 | 9.5% |
| BJ | 1.0% | 4.8% | 0 | 21.1% | 9.5% |
| BF | 4.0% | 6.0% | 11.8% | 78.9% | 23.8% |
| PC | 3% | 1.2% | 5.9% | 0 | 0 |

Table 7.7: Distribution of different intended message categories in different sources. *The Wall Street Journal* is shown as *WSJ*.

Input to the Intention Recognition Module is the augmented XML representation of a graphic produced by the Graph Segmentation Module. We used leave-one-out cross

validation in which each of the graphics is used once as the test graphic, with the conditional probability tables computed from the other 239 graphics. Our system recognized the correct intended message with the correct parameters for 173 line graphs, which gave us a 72.08% overall success rate. The accuracy rate for each intended message category is listed in Table 7.8. Failure of the Suggestion Generation Module to generate a candidate message matching the correct intended message prevents our Bayesian network from recognizing the correct intended message. If we compute the success rate on just the 215 line graphs for which the candidates produced by the Suggestion Generation Module included the intended message, the system achieves an 80.4% success rate.

| Intended message | # of training cases | Percentage | Accuracy |
|---|---|---|---|
| RT | 66 | 27.5% | 93.94% |
| FT | 22 | 9.17% | 86.36% |
| ST | 1 | 0.42% | 0% |
| CT | 58 | 24.17% | 79.31% |
| CTR | 22 | 9.17% | 40.91% |
| CTLS | 17 | 7.08% | 52.94% |
| CSCT | 7 | 2.92% | 28.57% |
| BJ | 11 | 4.58% | 36.36% |
| BF | 31 | 12.92% | 61.29% |
| PC | 5 | 2.08% | 60% |
| Total line graphs | 240 | 100% | 72.08% |

Table 7.8: Distribution of intended message in training data and the corresponding accuracy

If we use the message category that occurs most often as a baseline, namely Rising-Trend at 27.5%, then our system's success rate represents an improvement of 162.1% over merely selecting the most prevalent message category. However, it should be noted that our system not only must identify the correct message category but also must accurately identify the parameters of the intended message.

The system's errors are partially due to sparseness of data. For example the worst result is produced for the Stable-Trend and Contrast-Segment-Change-Trend intended message categories; we only have 1 and 7 instances of these two categories respectively.

When we do a leave-one-out cross validation for the Contrast-Segment-Change-Trend intended message category, we only have the remaining 6 training cases to contribute to the conditional probability tables. If some feature occurs in 2 of the 7 instances, and we use one of them as a test case, the conditional probability in the evidence node reflecting the occurrence of that feature drops significantly from 28.57% to 16.67%. Similarly if we have only one graphic where a particular verb class is used to indicate an intention category, then leave-one-out cross validation has no means to connect the verb class with that intention category and we are likely to get an incorrect result when hypothesizing the intended message of that graphic.

In addition, if the Graph Segmentation Module does not produce the correct segmentation of a graphic, the Suggestion Generation Module is unlikely to produce a set of suggested messages that includes the graphic's intended message, and thus the Bayesian network will not correctly hypothesize it. From this perspective, the Graph Segmentation Module places an upper bound of 89.58% on the system performance.

However, it should be noted that for some cases even when our system does not produce the ideal result, the message hypothesized by our system still reflects the information in the graphic if the trend segments provided by our Graph Segmentation Module are correct.

## 7.5 Identifying sub-intended message category for Big-Jump and Big-Fall intended messages

In our corpus we have noticed that the graph designer usually follows a sudden sharp rising or falling segment with a series of data following these sudden changes. We hypothesize that this series of data are used by the graph designer to show what happens following the sudden big jump or big fall. For example, Figure 7.8a conveys a message that there was a big jump in the number of lay-offs from the second quarter of 2000 to the third quarter of 2001; it also conveys a message that after the big jump in the number of lay-offs ended in the third quarter of 2001, the number of workers laid off didn't stay at the

| Name of Evidence | Description and values |
|---|---|
| SustainSlope | Measures the slope of the regression line of the data points following the sudden rising or falling segment. It is a continuous number. |
| SustainThreshold | Measures the ratio between two heights: the height between the lowest and the highest points in the data series following the Big-Jump/Big-Fall segment, and the height of the Big-Jump/Big-Fall segment. This ratio is a continuous value between $0$ and $1$. |

Table 7.9: The evidence nodes and their corresponding values for sub-intended messages Big-Jump-Sustain/Big-Jump-NotSustain and Big-Fall-Sustain/Big-Fall-NotSustain

high level but dropped until the first quarter of 2006. We refer to this failure to sustain the big jump as a sub-intended message. In our training data, we observe that only Big-Jump and Big-Fall intended messages have apparent sub-intended messages which are either "Sustain" or "Not Sustain", reflecting whether or not the values remain at the high/low level until the end of the graphic. So the four sub-intended message categories are Big-Jump-Sustain/Big-Jump-NotSustain and Big-Fall-Sustain/Big-Fall-NotSustain. The parameters of these sub-intended messages are the same as the parameters for the Big-Jump or Big-Fall intended message, which are the start point and end point of the sudden rising or falling segment.

For each training line graph with Big-Jump or Big-Fall intended message, the human annotators discussed and reached a consensus on its sub-intended message. To build a model to identify the sub-intended message, we use the C4.5 decision tree algorithm[76] provided by Weka[94]. The evidence passed to the decision tree includes the two listed in Table 7.9 and the "XDuration", "YScale", "LastLength", "TouchEnd", "BigSlope", "SmallLargeRatio" and "LastSegmentMatchWord" shown in Table 7.2 and 7.3. The training set contains 42 training instances with Big-Jump and Big-Fall intended messages, 24 of them have "Sustain" sub-intended messages and 17 of them have "Not Sustain" sub-intended messages. Using leave-one-out cross validation on this training set, the decision tree achieves a $97.6\%$ (41 out of 42) success rate at identifying whether or not the sudden

change is sustained, which has a 70.9% improvement over the baseline method which achieves a 57.1% success rate by always choosing the most frequent sub-intended message which is "Sustain" in our training set.

## 7.6 Summary

This chapter presented our methodology for recognizing the intended message of a line graph. It constructed a probabilistic graphical model that arbitrates among the message candidates suggested by the Suggestion Generation Module. The communicative signals are used as evidence nodes in the Bayesian network. Our system achieves a success rate of 72.08% on 240 training instances using leave-one-out cross validation. Chapter 9 presents a demonstration of the intended message recognition system used as part of an assistive technology project for individuals with sight impairments.

# Chapter 8

# MOST RELEVANT PARAGRAPH IDENTIFICATION

Our work on recognizing the intended message of a line graph has been discussed in the preceding chapters. The system for recognizing the intended messages can contribute to several areas such as assistive technology, multimodal document summarization and retrieval of information graphics. However, we hypothesize that for each of these applications, identifying the paragraphs in a multimodal document that are most relevant to the information graphic will be necessary. The identification of the most relevant paragraph relates the intended message of an information graphic to a sub-part of its enclosing article. The rest of this chapter is outlined as follows. Section 8.1 discusses the necessity and potential contribution of identifying the most relevant paragraph for several research areas. Section 8.2.1 discusses our basic method for identifying the most relevant paragraph – KL divergence. It is not only a text similarity measurement but also has background from generative probability. Section 8.2.2 and 8.2.3 discuss our two methods for learning an expansion word list to improve the result. Section 8.3 evaluates our method and compares it with another method. Section 8.4 presents two examples illustrating the effectiveness of our method. Section 8.5 discusses related work from research areas such as passage retrieval and question answering. The conclusion summarizing this whole chapter and the future work are in Section 8.6.

## 8.1 Importance of relevant paragraph identification

Unlike scientific articles, the texts of multimodal documents from popular media rarely refer explicitly to their information graphics and the graphics often do not appear

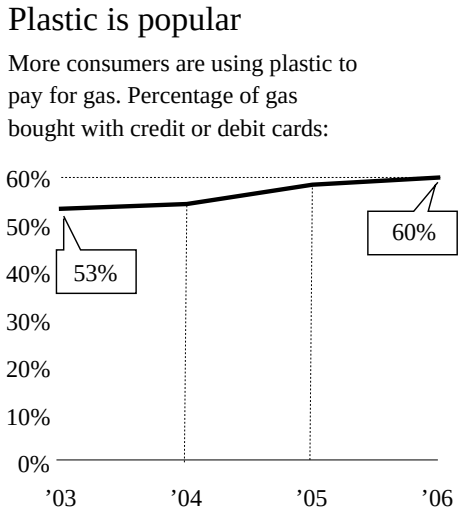



Figure 8.1: A line graph from an article about consumer spending where the most geographically adjacent paragraph is not relevant to the line graph

adjacent to a relevant paragraph (or even on the same page).

For example, the graph in Figure 8.1 is included in an article published in *USA Today* with the headline *"Paper or plastic? Answer might save at the pump"*. The most relevant paragraph within the article is the following:

- *"More than three-quarters of the gas pumped in the USA is sold at convenience stores. In 2005, 58% of gas was bought using credit and debit cards. Retailers say that number has been climbing in 2006, Lenard says."*

But the paragraph closest to the line graph is the following, which is not relevant to the line graph:

- *"But on a recent Monday morning, the restaurant owner from Edgemoor, S.C., took out his wallet, went into the gas station convenience store and paid with cash to take advantage of a 4-cent discount for cash customers."*

Section 8.1.1, 8.1.2 and 8.1.3 discuss three projects where we hypothesize that it would be beneficial to identify the paragraphs most relevant to an information graphic.

### 8.1.1 Assistive technology

The intended message recognition system can be used in an assistive technology project to provide blind users with access to the high-level content of an information graphic. Once the intended message of an information graphic has been recognized, it can be realized as English text and the screen-reading software JAWS can read the brief textual summary to the user. This is the approach taken by the SIGHT system which is described in the next chapter. One option is for the SIGHT system to prompt the user at the location where the information graphic appears in the webpage. But as Section 8.3 will show, paragraphs that are geographically closest to where the information graphic is displayed are often not relevant to the information graphic. Practically, if the screen-reading software is reading some content and then suddenly mentions that there is a line graph, blind users have no clue about whether it is worth the time to listen to the summarization of the information graphic since it might be totally irrelevant to what has been read in the text. Moreover, summarizing the graphic at the wrong location will likely interrupt the user's understanding of the whole article rather than assisting it. On the other hand, when the screen-reading software is reading some content which is relevant to a line graph in the document, if the software doesn't mention there is an information graphic relevant to the current content, the blind user will miss it and thus have an incomplete understanding of this portion of the multimodal document. Therefore, ideally the user should be prompted about the existence of an information graphic at the location in the text that is most relevant to the graphic's content, so that the overall reading of the document(both the content of the information graphic and the text of the article) is coherent.

### 8.1.2 Summarization of multimodal documents

Although abstractive summarization is the Holy Grail of summarization research, the state-of-the-art is extractive summarization in which important clauses or sentences are extracted from a document's text. The extracted text is then knitted together into a

summary, with the pieces of text generally appearing in the same order as in the original article.

Extractive summarization research has focused on text, and little attention has been given to multimodal documents. For the most part, this has been due to the difficulty of identifying the content of non-textual components of a document and how this content relates to the document's text. As shown by [11], the message conveyed by an information graphic in popular media (such as newspapers and magazines, as opposed to scientific articles) is often not repeated in the article's text; furthermore, the graphic's caption often contains little or none of the graphic's primary intended message. Thus, information graphics in multimodal documents cannot be ignored.

We hypothesize that our work on intended message recognition might be applied to the summarization of multimodal documents by inserting the graph's summary into the document's text and then applying traditional extractive summarization techniques to construct a summary of the entire document. However, the graph's summary must be inserted at a relevant point in the document if extractive summarization techniques are to succeed. Thus extractive summarization of a multimodal document requires that the appropriate placement of content from its information graphics be identified.

### 8.1.3 Retrieval of information graphics

Information graphics contain knowledge that is often not repeated in the article's text. Thus a system that could retrieve information graphics from a digital library would be very useful. Current retrieval systems do not try to understand, and take into account, the graphic's high-level content. We hypothesize that a good approach to graph retrieval is a mixture model. This mixture model should take into account the graphic's intended message (recognized by our system) and the textual component of the graphic which is its caption/description/text-in-graphic[1], and the article containing the information graphic.

[1] introduced in Chapter 5.

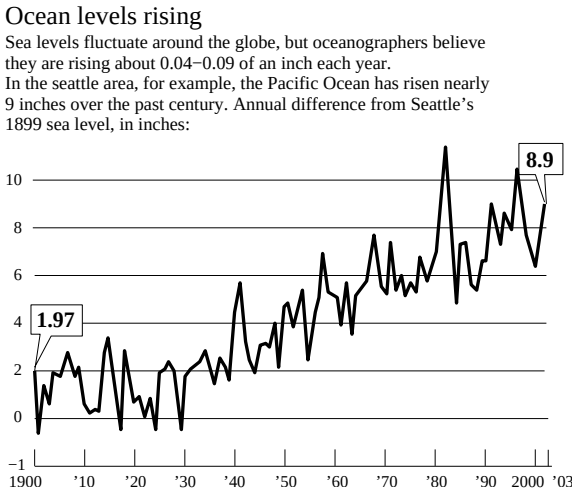



Figure 8.2: A line graph which appears in an article with multiple topics

However, the article may contain multiple topics, many of which are not relevant to the information graphic. For example, an article in popular media discussing global warming contains Figure 8.2 showing a changing trend in ocean levels. But it also discusses other topics such as how an animal's living habits have changed and how the weather has changed. Those topics are not relevant to Figure 8.2, and thus we hypothesize that these portions of the text should not be considered in the mixture model for deciding whether to retrieve this graphic in response to a user query. Therefore, the mixture model using the whole article may not be appropriate. If we can identify several paragraphs relevant to an information graphic and use only them in the mixture model, we hypothesize that the results are likely to be better than if the whole article was used.

## 8.2 Identifying the most relevant paragraph

To identify the paragraphs that are most relevant to an information graphic, we will take advantage of the textual component in an information graphic and the instantiated parameters of the graphics intended message. The methods discussed in this chapter have been applied to line graphs. The textual component of a line graph consists of caption,

description and text-in-graphic[2].

Section 8.2.1 proposes a KL divergence based calculation which measures the similarity between the textual component of the line graph and the paragraphs. Section 8.2.2 then proposes a second method that augments the textual component with words selected from a word list consisting of verbs and adjectives that commonly appear in multimodal documents, and with the parameters of the intended message of a line graph. The first part of the expansion word list reflects domain-independent graphic content and thus captures words that might appear in a paragraph relevant to *any* information graphic; the parameters of the intended message reflect the line graph's specific content and thus might appear in a paragraph that is specific to this information graphic. Section 8.2.2.1 introduces our method to learn the expansion word list. Section 8.2.3 further extends the method discussed in Section 8.2.2 by applying the KL divergence with expansion word list to measure the relevance of both a paragraph and each sentence in this paragraph. The linear combination of the scores from a paragraph and the most relevant sentence in this paragraph is used to rank the paragraph.

In our current work, we assume that a document contains a single information graphic; future work will extend our methodology to documents with multiple information graphics or composite information graphics.

### 8.2.1 Method P-KL: KL divergence

Our basic algorithm uses Kullback-Leibler divergence to measure the similarity of two language models, one model for a paragraph in a document and the other model for the information graphic's textual component. KL divergence has been widely used in natural language processing and text mining. It measures the difference between two distributions, either continuous or discrete, and can be written as

$$D_{KL}(p||q) = \sum_{i \in V} p(i) log \frac{p(i)}{q(i)} \tag{8.1}$$

[2]introduced in Chapter 5

where $i$ is the index of a word in vocabulary V, and $p$ and $q$ are two distributions of words. If $p$ and $q$ represent the same word distribution, $D_{KL}(p||q)$ will be $0$. For our problem of identifying the relevant paragraphs, $p$ is a smoothed word distribution built from the line graph's textual component, and $q$ is another smoothed word distribution built from a paragraph in the corresponding document. Smoothing addresses the problem of instances with zero occurrences of a word in the word distribution, which will cause problems in computing the KL divergence. We assign the observed word its true word frequency and assign each unobserved word a low frequency (such as 0.01) and then normalize the word distribution. We rank the paragraphs by their KL divergence score from lowest to highest, since lower KL divergence scores indicate a higher similarity.

Although KL divergence and cosine similarity both seem to be distance measurements, KL divergence has more meaning for our purposes than cosine similarity. We can rewrite Equation 8.1 as follows:

$$
\begin{aligned}
D_{KL}(p||q) &= \sum_{i \in V} p(i) log \frac{p(i)}{q(i)} && (8.2)\\
&= \sum_{i \in V} p(i) log(p(i)) - \sum_{i \in V} p(i) log(q(i)) && (8.3)\\
&= \sum_{i \in V} p(i) log(p(i)) - \sum_{i \in V} \frac{c_{p,i}}{\sum_{i \in V} c_{p,i}} log(q(i)) && (8.4)\\
&= \sum_{i \in V} p(i) log(p(i)) - \frac{1}{\sum_{i \in V} c_{p,i}} \sum_{i \in V} c_{p,i} log(q(i)) && (8.5)\\
&= \sum_{i \in V} p(i) log(p(i)) - \frac{1}{\sum_{i \in V} c_{p,i}} log \prod_{i \in V} q(i)^{c_{p,i}} && (8.6)\\
& && (8.7)
\end{aligned}
$$

where we assume that the word probability in distribution $p$ is obtained by word counting as $\frac{c_{p,i}}{\sum_{i \in V} c_{p,i}}$ where $c_{p,i}$ is the count of word $i$ in the textual component of a line graph to obtain word distribution $p$, and the $q(i)$ is the probability of generating word $i$ from a paragraph. The rank of a paragraph whose word distribution is represented by $q$ is

determined by $\prod_{i \in V} q(i)^{c_{p,i}}$ which is the probability of generating the textual component of the graphic from a paragraph. This model matches our intuition well since usually the words in an information graphic are much fewer than in paragraphs.

### 8.2.2 Method P-KLE: KL divergence with augmented textual component

In our KL divergence calculation, measuring the distribution distance from the textual component of a graphic to a paragraph is essentially measuring the generative probability of the textual component from a paragraph. But the number of words in the textual component of an information graphic is small. This will increase the sparsity problem since some words in the paragraph may be relevant to the information graphic but are not contained in the textual component of the graphic. Moreover, the textual component can vary depending on the domain, whereas much of the actual graphic has domain-independent information. For example, most line graphs present trends, rises or falls, results(higher or lower), and some bar charts represent ranks or comparisons. This sparsity problem and loss of domain-independent information in the textual component suggests a potential improvement for the identification of the most relevant paragraphs. Thus we decided to explore whether we could automatically extract a set of expansion words that are commonly used in paragraphs that are relevant to information graphics. In the following discussion, we refer to it as the "expansion word list".

The expansion word list is identified using a relevance feedback technique. It is an iterative process where pseudo-relevant paragraphs are first selected using KL divergence on each graphic's textual component, and a set of words relevant to the textual component are identified. In each subsequent iteration, for every information graphic, we extract $k$ pseudo-relevant paragraphs using KL divergence with the textual component augmented with the expansion word list produced on the previous iteration. Therefore at the end of an iteration, there will be at most $k \cdot N$ pseudo-relevant paragraphs selected where the total number of articles is $N$.

The word list resulting from each iteration is a ranked list of words. We assume that the ideal words are verbs and adjectives because they have minimal binding with the domain-specific knowledge of each information graphic. For example, in our data set collected from popular media, there might be many information graphics conveying financial information where the relevant paragraphs contain words such as "share", "price", etc. But we shouldn't arbitrarily extend an information graphic's textual component with these words since they are not relevant to information graphics in other domains. Those words are too domain-specific compared with words like "rise", "drop", "up", etc. Therefore we filter the resulting word list by using WordNet to remove the words whose dominant sense is neither *verb* nor *adjective*. To determine the dominant sense from WordNet, we calculate the number of senses belonging to different categories such as *noun*, *verb*, *adjective*, etc. For example, the word "rise" has 10 noun senses and 17 verb senses, so the word "rise" is regarded as a *verb* and won't be filtered. Since the filtered word list is also a ranked list of all observed verbs and adjectives, we choose the top $l$ words from the word list. In our experiments, we chose $l = 25$.

In addition, the parameters of an intended message capture domain-specific content of the graphic's communicative goal. For example, the intended message of the line graph in Figure 8.2 is ChangeTrend(1900, stable, 1930, rise, 2003) conveying a changing trend in ocean levels over the period from 1900 to 2003 with the change from relatively stable to rising occurring in 1930. The parameters 1900, 1930 and 2003 may not appear in the graphic's textual component yet may appear in a relevant paragraph. Thus we also added the parameters of the intended message to the expansion word list.

We assume in each iteration that the expansion word list will be improved, (only the machine learned part of the expansion word list will be improved, whereas the parameters of the intended messages of the information graphics will stay the same between iterations), and thus the pseudo-relevant paragraphs will be closer to the actual relevant paragraphs, which facilitates obtaining a better expansion word list on the next iteration.

These iterations continue until the learned expansion word list stays the same or changes minimally.[3]

To select the most relevant paragraph in a new test document containing an information graphic, we apply KL divergence to relate the textual component of the graphic augmented with the expansion word list to each paragraph of the document. Because the textual component may be even shorter than the expansion word list, we don't add a word from the expansion word list to the textual component unless the compared paragraph also contains this word.

#### 8.2.2.1 Learning the expansion word list from word frequency

To learn the expansion word list, we regard the articles and pseudo-relevant paragraphs as two sets of words, i.e. we disregard from which document a pseudo-relevant paragraph was selected. We assume that the collection of pseudo relevant paragraphs was generated by two independent models, one producing words relevant to the information graphics and one producing words relevant to the topics of the documents. Let $W_g$ represent the word frequency vector that generates words relevant to the information graphics, $W_a$ represent the word frequency vector that generates words relevant to the set of articles, and $W_p$ represent the word frequency vector of the set of pseudo-relevant paragraphs.

We can compute $W_p$ from the whole set of pseudo-relevant paragraphs, and we can estimate $W_a$ as the word frequency vector for the entire set of articles. We want to compute $W_g$ by filtering the components of $W_a$ from $W_p$. This is similar to the work done by Widdows[93] on orthogonal negation of vector spaces. The problem can be formulated as follows:

1. $W_p = \alpha W_a + \beta W_g$ where $\alpha > 0$ and $\beta > 0$, which means the word frequency vector for the pseudo-relevant paragraphs is a linear combination of the background (topics) word frequency vector and the graphic word vector.

[3]There is no proof that the pseudo-relevant paragraphs and the expansion word list will converge.

2. $< W_a, W_g >= 0$ which means the background word vector is orthogonal to the graph description word vector. We assume that when the author writes paragraphs that are unrelated to the graphic, he/she will not have the graphic words in mind. Therefore the graphic word vector is independent of the background word vector and these two share minimal information. Since we use a vector space model to represent $W_a$ and $W_g$, orthogonality is obtained by assuming that these two word vectors have minimum similarity.

3. $W_g$ is assumed to be a unit vector. Whether or not $W_g$ is a unit vector is immaterial for our method, since we are interested only in the relative rank of the word frequencies, not their actual values. However, assuming that $W_g$ is a unit vector gives us three equations in three unknowns ($W_g$, $\alpha$, and $\beta$) which can be solved for $W_g$.

With these three assumptions, we obtain

$$\alpha = \frac{< W_p, W_a >}{< W_a, W_a >} \tag{8.8}$$

$$W_g = \text{normalized}\left(W_p - \frac{< W_p, W_a >}{< W_a, W_a >} \cdot W_a\right) \tag{8.9}$$

The calculated $W_g$ is used in our methods P-KLE and P-KLEM. The top $k$ words which have the highest word frequency are selected as the expansion word list.

### 8.2.3 Method P-KLEM: using sentence in addition to paragraph to improve the result

Sometimes we consider a paragraph relevant only because there is a relevant sentence in the paragraph, without contribution from other sentences. We hypothesize that taking into consideration both the best sentence in a paragraph and the paragraph itself may further improve the result. We implement another method named P-KLEM, which computes the final score for a paragraph as a weighted sum of the original score for the

paragraph and the score for the best sentence in the paragraph (the sentence with the lowest KL divergence from the augmented textual component).

$$Score_{final_p} = \lambda Score_{\text{best sentence}_{\in p}} + (1 - \lambda) Score_p$$

In our experiment, we arbitrarily choose $\lambda = 0.5$.

## 8.3 Evaluation

In the following evaluations and examples, we will denote the KL divergence method without the expansion word list as P-KL where "P" represents the paragraph selection and "KL" means that we are using KL divergence to select the most relevant paragraphs. We will denote the method using the expansion word list as P-KLE, where "P-KL" means we are using KL divergence to select paragraphs, and "E" means the textual component of a line graph is augmented by the expansion word list. The further improvement (which combines the scores from a paragraph and the most relevant sentence in the paragraph as discussed in Section 8.2.3) is represented as "P-KLEM", where the "E" means that it uses the expansion word list and "M" means that it uses a mixture of scores from both paragraph and sentence.

### 8.3.1 The dataset

We have compiled a dataset of 367 information graphics with full articles from multiple national sources such as *USA Today*, *Businessweek*, *Newsweek*, *The New York Times*, *The Wall Street Journal* and some local sources such as *The Wilmington News Journal*. 100 graphs and articles out of the 367 information graphics had been analyzed by two human evaluators. For these articles, the two human evaluators identified paragraphs in each document that were relevant to its constituent information graphic and ranked them in terms of relevance. On average, Evaluator-1 selected 1.99 paragraphs and Evaluator-2 selected 1.67 paragraphs. For 66% of the graphs the two evaluators agreed on

the top ranked paragraph, with a kappa statistic of 0.624[4]; the other 34% of the graphics show that in many cases, the most relevant paragraph is not obvious and several possibilities exist. We held out the 100 line graphs as a test set and used the remaining line graphs to learn the expansion word list.

### 8.3.2 Evaluation criteria

After the expansion word list was learned from the set of 267 information graphics with accompanying articles, all three of our methods (P-KL, P-KLE and P-KLEM) processed the 100 test graphics with accompanying articles, and each method produced a ranked list of the paragraphs in terms of relevance. We evaluated the results in several ways. For summarization, we want to insert the summary of the graphic at a coherent point in the article's text and then apply extractive summarization on the text. For the assistive technology project for blind users, we want to prompt the blind users about the existence of an information graphic at the most relevant paragraph. These require us to use only the top result of our ranked list of relevant paragraphs for each document and thus lead to two evaluation criteria:

1. TOP: the method's success rate in selecting *the most relevant paragraph*, measured as how often the most relevant paragraph identified by the method matches one of the two evaluator's top-ranked paragraph.

2. COVERED: the method's success rate in selecting *a relevant paragraph*, measured as how often the most relevant paragraph identified by the method matches one of the paragraphs identified as relevant by the evaluators.

For our work on retrieving information graphics from a digital library, we want to use several paragraphs in the accompanying article to replace the article component

[4]Since the selection of paragraphs is different for each subject, the probability of chance agreement in kappa statistic is computed assuming that the probability of selection is 1/n where n is the number of paragraphs in the document.

in our mixture model[104, 42, 3, 65] to rank graphics for retrieval. Therefore we want to select several top ranked paragraphs. Thus an appropriate evaluation criteria is normalized discounted cumulative gain ($nDCG$)[44, 19]. The $nDCG$ is between $0$ and $1$, and measures how well the rank-order of the paragraphs retrieved by our method agree with the rank-order of the paragraphs identified as relevant by our evaluators. $nDCG$ is defined by the following formulas:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{8.10}$$

$$\text{where } DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{8.11}$$

$$\text{and } IDCG_p \text{ is the highest possible } DCG_p \tag{8.12}$$

The $rel_i$ is the gain of retrieving a paragraph and the $\frac{1}{log_2(i)}$ is the discount according to its position $i$. The $DCG$ calculation used in Information Retrieval as shown in Equation 8.11 doesn't assign a discount to the document at rank $2$ (because $log_2(2) = 1$). Thus following Burges[6, 7, 8], we have changed the calculation of $nDCG$ by assigning discount $log_2(i+1)$ to rank $i > 1$ in Equation 8.11 instead of using $log_2(i)$. Our $nDCG$ is measured as follows:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{8.13}$$

$$\text{where } DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \tag{8.14}$$

$$\text{and } IDCG_p \text{ is the highest possible } DCG_p \tag{8.15}$$

We set the cut off position at $p = 3$. The value of $rel_i$ depends on $p$ and the number of relevant paragraphs identified by the human evaluator. If the human evaluator identifies $k$ paragraphs as relevant (where $k \leq p$), then $rel_i = k$ if the i-th ranked paragraph by the system matches the top-ranked paragraph by the human evaluator and is equal to $k - 1$ or $k - 2$ if it matches the paragraph ranked second or third by the human evaluator,

Table 8.1: Machine learned expansion word list

| | |
|---|---|
| P-KLE | up down rise according fall high expect low late grow decline hit federal buy free drop worry dip long august jump risky back average surge |

respectively. Ranking a good paragraph higher gets less discount with the same gain, and ranking a better paragraph at the same position gets higher gain with the same discount. They both achieve a better nDCG score.

### 8.3.3 Experimental results

First let's take a look at the expansion word list learned by P-KLE. The top 25 words in the ranked word lists provided by P-KLE is shown in Table 8.1. Although there is still noise in the word list after the pseudo relevance feedback with filtering and iterations, most words, such as "up","rise","down","fall", are relevant to line graphs. We observe that the word "according" is ranked high. We suggest that authors who want to include data professionally add in the data source as "according to XXX" and then add an information graphic to convey their message about the data. Thus the phrase "according to" co-occurs often with line graphs. The word "average" is also ranked fairly high. Although it doesn't describe trends, it may be used to describe data shown in the information graphics. The word "hit" is used interestingly in popular media when mentioning outstanding data, as in "sales hit $106 million" or "The September unemployment rate hit 25.9%, the highest rate since World War II". Thus the expansion word list contains words that are independent of the domain but are often used in paragraphs relevant to information graphics.

Figures 8.3 and 8.4 and Table 8.2 present the success rate for all of our methods for criteria TOP and COVERED, along with the success rates for two baseline methods: 1) selection of a random paragraph as most relevant (labelled "random"), and 2) selection of the paragraph that is closest to the information graphic (labelled "nearby"). The results displayed in Figures 8.3 and 8.4 show that all of our methods based on the KL divergence

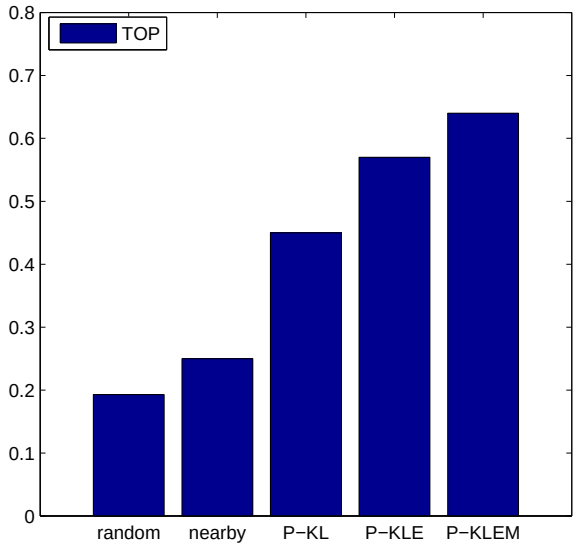



Figure 8.3: Success rate in selecting the paragraph identified as most relevant by one of the two human evaluators

(or generative probability) outperform the baseline methods. P-KLE with expansion word list is a further improvement on P-KL. The P-KLEM has the best result. P-KLE and P-KLEM select the best paragraph in 57% and 64% of the test cases respectively, and select a relevant paragraph in 67% and 78% of the cases, respectively. Techniques based on P-KL with expansion word list doubles or almost doubles the success rate of the baseline methods. The improvement of P-KLE and P-KLEM over P-KL indicates that our expansion word list successfully expands the textual component with words pertinent to the graphic itself.

| Criteria | random | nearby | P-KL | P-KLE | P-KLEM |
|---|---|---|---|---|---|
| TOP | 19.28% | 25% | 45% | 57% | 64% |
| COVERED | 23.14% | 37% | 62% | 67% | 78% |

Table 8.2: Success rate of each method for criteria “TOP” and “COVERED”

A binomial test assumes each test case generates a binary result, either “correct” or “incorrect” where “correct” means that the returned top result by the corresponding

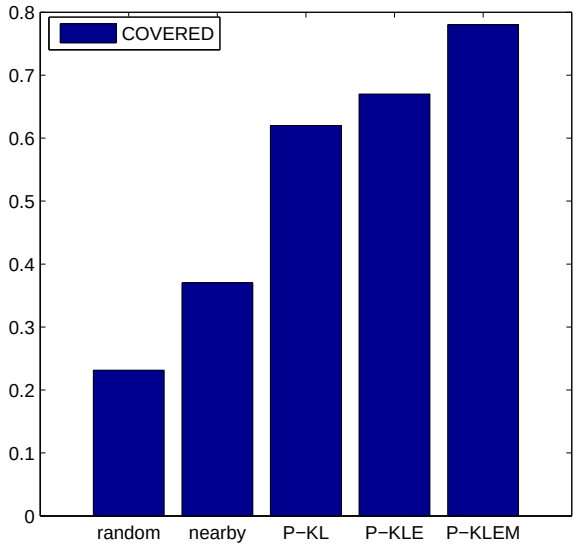



Figure 8.4: Success rate in selecting a paragraph identified as relevant by one of the two human evaluators

method matches the paragraph selected as best by one of the human evaluators under the "TOP" criteria, or matches one of the paragraphs identified as relevant by a human evaluator under the "COVERED" criteria. The null hypothesis is that the two compared success rates are the same. The binomial distribution can be approximated by a normal distribution because we have 100 test cases and the success probability used in binomial test is not near 0 or 1. The approximation using the normal distribution for the binomial test can be represented as

$$z = \frac{p - \hat{p}}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}})}$$

where $p$ is the success rate of the improved method and $\hat{p}$ is the success rate of the method it is being compared to.

The statistics presented in Table 8.3 show that our P-KL model has a significant improvement over just selecting the closest paragraph and P-KLE also has significant improvement over P-KL on the "TOP" criteria. P-KLEM provides significant improvements on both P-KL and P-KLE for both "TOP" and "COVERED" criteria.

| Comparison | TOP | | COVERED | |
|---|---|---|---|---|
| | Z value | significance level | Z value | significance level |
| P-KL over nearby | 4.6188 | 0.001 | 5.1781 | 0.001 |
| P-KLE over P-KL | 2.4121 | 0.01 | 1.0301 | not significant |
| P-KLEM over P-KL | 3.8191 | 0.001 | 3.2963 | 0.001 |
| P-KLEM over P-KLE | 1.4139 | 0.1 | 2.3394 | 0.01 |

Table 8.3: The Z value and one tail significance level when comparing the improvement between two methods

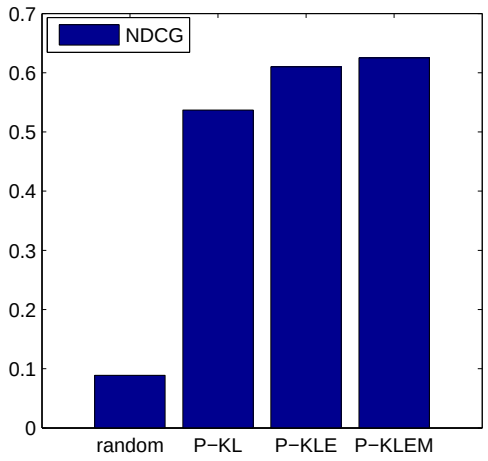



Figure 8.5: The nDCG scores provided by each algorithm, using random algorithm as baseline

For the criteria $nDCG$, Figure 8.5 presents the results of evaluating the methods in terms of the ranked order of their top three results. We measure $nDCG$ using each of the two evaluators as the ideal, and then average the results. (When comparing the two human evaluators against one another, their average $nDCG$ is 0.707.) The baseline method in this evaluation is a random selection of three paragraphs from each document. We use the *student t test* for two related samples to evaluate the results[37], which is calculated as

$$t = \frac{\bar{D} - 0}{s/\sqrt{n}}$$

where $\bar{D}$ is the mean of the differences between two $nDCG$ scores generated by two

| | *t score* | one tail significance level |
|---|---|---|
| P-KL over random | 7.95 | 0.001 |
| P-KLE over P-KL | 4.08 | 0.001 |
| P-KLEM over P-KL | 3.91 | 0.001 |

Table 8.4: The $t$ score and one tail significance level while comparing the different methods on $nDCG$

methods. (The null hypothesis is that the two sets of $nDCG$ scores do not differ, which is reflected by the $0$ in the numerator). Let $s$ be the sample standard deviation of the differences between the two sets of $nDCG$ scores calculated as $s = \sqrt{\frac{SS}{n-1}}$ where $SS$ is the sum of the squared differences. Let $n$ be the number of cases used; in our experiment, $n = 100$. The degree of freedom of this *t test* is $n - 1$. Table 8.4 presents the *t score* and the one-tail significance level using 99 as degree of freedom. The results in Figure 8.5 and Table 8.4 show that all of our methods outperformed the baseline. Methods P-KL, P-KLE and P-KLEM all more than quadrupled the $nDCG$ of the baseline method. The improvements of P-KLE and P-KLEM over P-KL are both statistically significant at the 0.001 significance level.

### 8.3.4 KL divergence versus cosine similarity

The cosine similarity has been widely used in early information retrieval research. It regards two bags of words as two word vectors and calculates the angle between these two vectors to measure their similarity. If the two vectors are represented as A and B, the cosine similarity is calculated as

$$\frac{\sum_{i=1}^{|V|}(A_i \cdot B_i)}{\sqrt{\sum_{i=1}^{|V|} A_i^2} \cdot \sqrt{\sum_{i=1}^{|V|} B_i^2}}$$

where $|V|$ is the total number of words in the dictionary, and $A_i$ and $B_i$ are the count of word $i$ in word vector $A$ and $B$ respectively. Although cosine similarity measures the similarity between two word vectors, it doesn't have as solid a statistical explanation as KL divergence. We performed an experiment comparing the results provided by cosine

similarity and KL divergence measurements for relevant paragraph identification. These are shown in Table 8.5 where "P-CS" means cosine similarity and "P-CSE" means cosine similarity with the expansion word list. Without the expansion word list, cosine similarity produces better results than KL divergence; however it doesn't improve much with the expansion word list. The improvement of P-KLE over P-KL is much greater than the improvement of P-CSE over P-CS, and P-KLE performs better than P-CSE on the "TOP" criteria and equally as well as P-CSE on the "COVERED" criteria. We hypothesize that because of the essence of the generative probability, KL divergence with the expansion word list can better address the sparsity problem and thus provide a bigger improvement.

| Criteria | P-CS | P-CSE | P-KL | P-KLE |
|---|---|---|---|---|
| TOP | 0.49 | 0.54 | 0.45 | 0.57 |
| COVERED | 0.65 | 0.67 | 0.62 | 0.67 |

Table 8.5: Comparing cosine similarity with KL divergence over two criteria

We have also analyzed the performance of KL divergence and cosine similarity on different document sizes defined according to the number of paragraphs. We separated the testing set into three groups: the first group (referred to as small size document group) contains 31 multimodal documents with between 2 and 10 paragraphs; the second group (referred to as middle size document group) contains 35 multimodal documents with between 11 and 15 paragraphs; the third group (referred to as large size document group) contains 34 multimodal documents with more than 15 paragraphs each. The average number of paragraphs are 6.65, 13.67, and 25.09 for the small, middle, and large size document groups respectively. The results for the criteria "TOP" and "COVERED" for both KL divergence and cosine similarity are shown in Table 8.6. We can see that for all groups and both similarity measurements, the expansion word list brings improvement. For small size documents, KL divergence performs better than cosine similarity but for large size documents the cosine similarity performs better than KL divergence. In our

corpus, although large size documents contain more paragraphs, they have fewer words in each paragraph. In our test set, the documents in the small, middle, and large size document groups have on average 50.3, 40.0 and 32.1 words in each paragraph, respectively. Because KL divergence is based on generative probability, the smaller paragraphs will bring more sparsity to the KL divergence calculation. Although we have used the expansion word list to address the sparsity problem within the textual component of a line graph, the sparsity problem brought by the paragraph wasn't solved by our method. Therefore, for the shorter paragraph in a large size document, the cosine similarity is hurt less than the KL divergence and thus cosine similarity produces better results. A further improvement on our method may be achieved by choosing between KL divergence and cosine similarity according to the size of the paragraph.

| Document Group | Criteria | P-CS | P-CSE | P-KL | P-KLE |
|---|---|---|---|---|---|
| Small | TOP | 0.548 | 0.548 | 0.613 | 0.677 |
| | COVERED | 0.613 | 0.645 | 0.645 | 0.677 |
| Middle | TOP | 0.514 | 0.543 | 0.487 | 0.571 |
| | COVERED | 0.714 | 0.714 | 0.743 | 0.743 |
| Large | TOP | 0.412 | 0.529 | 0.265 | 0.471 |
| | COVERED | 0.618 | 0.706 | 0.471 | 0.559 |

Table 8.6: Comparison between cosine similarity and KL divergence on the three document groups and two criteria

## 8.4 Examples

Consider first the graphic in Figure 8.1. It appeared in an article containing 38 paragraphs. The closest paragraph, which has little relevance to the graphic, is

> *"But on a recent Monday morning, the restaurant owner from Edgemoor, S.C., took out his wallet, went into the gas station convenience store and paid with cash to take advantage of a 4-cent discount for cash customers."*

The most relevant paragraph is repeated below:

> *"More than three-quarters of the gas pumped in the USA is sold at convenience stores. In 2005, 58% of gas was bought using credit and debit cards. Retailers say that number has been climbing in 2006, Lenard says."*

Both of our human evaluators selected this paragraph as most relevant to the graphic, and our best performing methods P-KLE and P-KLEM did the same.

Now consider the graphic in Figure 8.2. This graphic appeared in an article on global warming containing 23 paragraphs. Not only does the paragraph closest to the graphic have little relevance to it, but also no paragraph in the article stands out as overwhelmingly most relevant to the graphic. In fact, the two evaluators selected three and four paragraphs respectively as most relevant, and not only did they differ on their top-ranked paragraph but they also had only one paragraph in common. The top-ranked paragraph identified by all of our methods, P-KL, P-KLE and P-KLEM, are the same, which is

*"Rising sea levels are eroding beaches in the South Pacific."*

Although it does not match the paragraph identified as best by either of the human evaluators, the top four paragraphs selected by P-KLE and the top three paragraphs selected by P-KLEM include the four and three distinct paragraphs identified as relevant by one of the human evaluators respectively. In addition, the top three paragraphs selected by the P-KL method (which doesn't use an expansion word list) are also all relevant as annotated by the human evaluators. This performance on such a difficult article indicates that our method can handle articles where the most relevant paragraph is not obvious.

## 8.5 Related work

Our research on identifying the most relevant paragraphs is similar to existing work on passage retrieval[62, 60, 84, 107], and especially passage retrieval for question answering[16, 91, 61, 21, 9, 47]. Their work returns a window of text(can be a fixed length window or a paragraph) as the result of a query by either measuring the similarity between this textual window and the query or by using some heuristics such as using a passage containing "because" for a "why" query. To measure the similarity between the passage and the query, their work uses either dependency relation[21, 84] or other metrics

such as language models. In Khalid et al.'s work[47], they show that to answer a "why" question, KL (Kullback-Leibler) divergence provides better results than the *tfidf* metric which is usually used together with cosine similarity measurement.

Our research on identifying the most relevant paragraph differs from their work in two respects:

1. Different methods have been investigated and they usually use a fixed size window. However, because one application of our research is to provide a coherent location for the blind user to be prompted about the presence of an information graphic, the natural boundary of topic is more important than a fixed size window. Therefore, we chose to use natural paragraphs as the result of our method.

2. Our passage retrieval is limited to a single document. Therefore, the large scale relevance feedback techniques[58] used in the typical passage retrieval task which can retrieve thousands of passages from hundreds of documents for a single query are not applicable.

Pseudo relevance feedback techniques[101, 96, 97, 86, 10, 56, 17, 53] are also widely used in query expansion for text or image retrieval. Our research isn't trying to retrieve a large number of documents for a single query; instead we identify several paragraphs pseudo relevant with the textual component of an information graphic and use them to identify a set of words that are typically used in paragraphs relevant to information graphics.

Yu et al. [98] used a hierarchical clustering algorithm based on *tf-idf* to associate sentences from an abstract with images in biomedical articles. This is similar to our work since it is also trying to identify a segment of text which is most relevant to a figure. However in scientific documents, such as biomedical articles, figures are usually explicitly referred to by a sentence or paragraph. For example, a sentence may explicitly say "Figure 1 shows XXX" or "as shown in Figure 1, XXX". With scientific articles one

can locate the referring sentence first and this sentence will contain words relevant to the figure. These words can then be used to identify relevant sentences in the abstract. Information graphics in popular media generally have no number/labels such as "Figure 1" and no explicit reference from the text in the article. This makes our task more difficult.

A basic technique used in our work is Kullback-Leibler divergence. This has been investigated extensively in the work of Zhai[103, 51] and Lavrenko[54]. As shown in Section 8.2.1, our application of KL divergence can be reduced to a generative model which assumes the textual component of an information graphic is generated from a paragraph; this is similar to the generative model[51, 54] used in the information retrieval area which assumes that the queries are generated from documents. In the calculation of KL divergence, to avoid assigning zero probability to an unseen word, multiple smoothing methods[105, 106, 20] have been proposed and analyzed including Laplace smoothing, Jelinek-Mercer method and Bayesian smoothing using Dirichlet Priors. The smoothing used in our work is similar to the Laplace smoothing by assigning a fixed small count to all unseen words but keeping the count of seen words the same. This smoothing is very simple; replacing it by the more sophisticated smoothing techniques investigated by Zhai[105, 106] may improve performance.

## 8.6 Conclusion and future work

Identifying the most relevant paragraphs can address many problems that arise in using the intended message of an information graphic in several applications: an assistive technology project for blind users, the extractive summarization of multimodal documents, and the retrieval of information graphics from a digital library. This chapter has shown that the method based on KL divergence, which essentially measures the generative probability of the textual component of an information graphic from a paragraph, can produce good results either when we want to identify only the most relevant paragraph or multiple top paragraphs ranked by their relevance. The expansion word list, automatically learned from the dataset without any human intervention, can produce

further improvement on the basic KL divergence measurement. The addition of the expansion word list into the KL divergence measurement results in statistically significant improvements for all three evaluation criteria. Although these methods have been applied only to line graphs, they don't use specific features of line graphs, and so can be readily applied to other kinds of information graphics such as bar charts.

# Chapter 9

# THE SIGHT SYSTEM

Our methodology for recognizing the intended messages of line graphs has been integrated into the SIGHT system which provides blind users with access to the full content of multimodal documents. SIGHT[31, 32] is designed to work within Internet Explorer and uses JAWS screen-reading software. The text of the document is read to the user via JAWS. To provide access to the high-level content of an information graphic, SIGHT calls the Visual Extraction Module to provide an XML representation of the graphic, an inference module to identify the graphic's intended message which forms the basis for a brief summary of the graphic, and then FUF/SURGE to realize the logical representation of the intended message as English text which is then read to the user by the screen-reading software. SIGHT was originally implemented only for simple bar charts. To extend SIGHT to work for line graphs, several problems had to be addressed.

1. If the image is a line graph, the Visual Extraction Module produces an XML representation including all information about this line graph. This XML representation includes the elements of a line graph such as the caption, the description, the text-in-graphic and the tick-marks on the x axis and y axis. It also contains a sampling of the line graph. The Visual Extraction Module samples the line graph by capturing the coordinate of the pixels of two ends of each straight line. It might provide a set of sample points clustered at some location such as in Figure 9.1 because the VEM captures all straight segments even if the two end-points of a straight segment are

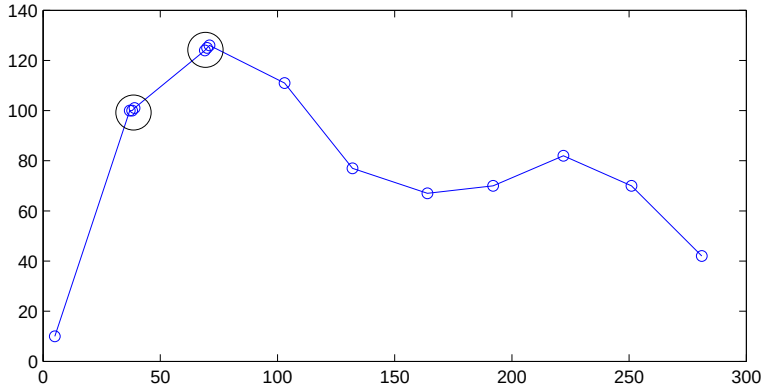


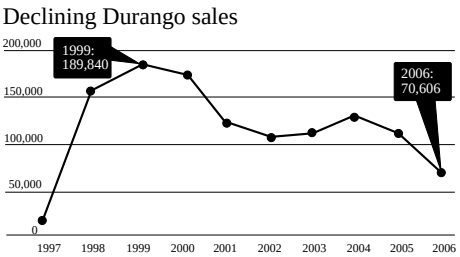

Figure 9.1: A plot of the sample points generated by VEM. The circled areas contain clusters of sample points because the VEM doesn't necessarily sample line graphs with uniform intervals.

very close (by one or two pixels). Therefore, the sample points don't have a relatively uniform distribution across the x axis as is required by the statistical tests used in the Graph Segmentation Module. Thus it was necessary to design a resampling module that provides a new set of sample points that balances the need for a uniform distribution against the need to include all of the outstanding points (the outstanding points are the local minimum or maximum points). The resampling procedure is described in Appendix A.

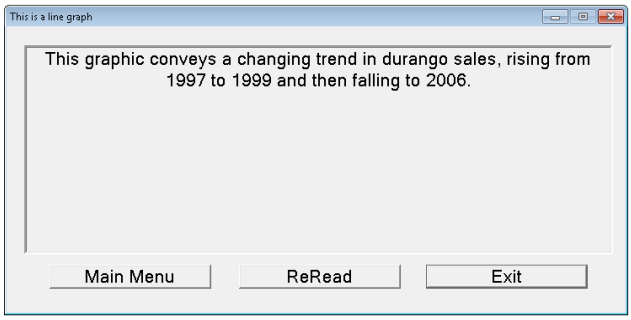
2. An inference mechanism had to be developed for recognizing the intended message of a line graph. This work was described in Chapter 5, 6, and 7.

3. In order to provide a more coherent presentation for the user, a methodology was needed for identifying the paragraph in a multimodal document that is most relevant to an information graphic. Chapter 8 presented our method for achieving this goal.

4. The above new modules had to be integrated into the SIGHT system and the overall processing had to be modified so that users are provided with access to line graphs




(a) A line graph showing a Change-Trend message




(b) A window in the SIGHT system shows the textual summary for Figure 9.2a

Figure 9.2: The brief summary of a line graph in SIGHT system

at the most relevant point in a document instead of where the information graphics appear in the document.

Note that the logical representation of the intended message produced by the inference module does not contain a referent to what is being measured – what we refer to as the measurement axis descriptor. For example, the measurement axis descriptors for Figure 9.2a and Figure 9.3a are *"Durango sales"* and *"annual difference from Seattle's 1899 sea level, in inches"* respectively. Seniz Demir[23] designed a module to extract the measurement axis descriptor by applying a set of heuristics to the various textual components of a line graph, such as the caption, description, text-in-graphic, etc. The logical representation of the intended message of a line graph along with the measurement axis descriptor are sent to a realization module that uses FUF/SURGE to produce an English summary.

## 9.1 Examples

The following two examples illustrate how the extended SIGHT system processes a multimodal document containing a line graph. Our first example is an article from *the Wilmington News Journal* containing the line graph in Figure 9.2a. Input to the implemented version of SIGHT is a webpage containing the xfig redrawn graphic and the

associated article. In the earlier version of the SIGHT system for bar charts, the screen-reading software prompted the user about the presence of a bar chart at the location where it appeared in the document. But as we saw in Chapter 8, the geographically closest paragraph might not be a coherent place to mention the presence of an information graphic. Based on our research on identifying the most relevant paragraph, the enhanced SIGHT system prompts the user at the paragraph most relevant to the graphic in Figure 9.2a, namely:

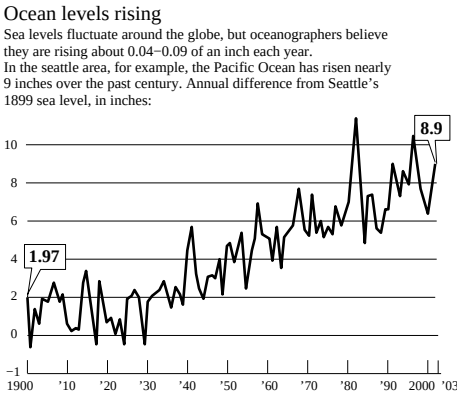
*Doing so likely would require the company to bring in a new model. Sales of the Durango and other gas-guzzling SUVs have slumped in recent years as prices at the pump spiked.*

If the user requests access to the graphic by typing Ctrl-Z, the Visual Extraction Module produces an XML representation of the line graph and the Caption Tagging Module identifies helpful words in the caption and description. The Resampling Module described in Appendix A is invoked to produce a relatively uniform distribution of sample points while still capturing change points in the data. Then the Message Recognition Module hypothesizes the graphic's intended message. The intended message of the line graph shown in Figure 9.2a is correctly identified and the textual summary "*This graphic conveys a changing trend in durango sales, rising from 1997 to 1999 and then falling to 2006.*" is produced by FUF/SURGE and read to the user by JAWS.

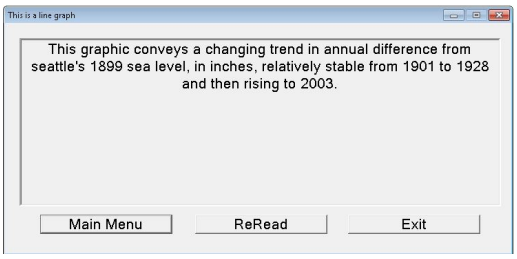
Another example illustrating the SIGHT system's performance is an article from *USA Today* containing the graphic shown in Figure 9.3a. In this document, the most relevant paragraph is not clear. Two human evaluators returned six paragraphs in total as relevant and they only agreed on one paragraph(not the top one of either of their ranked paragraphs). The SIGHT system prompts the user about the presence of this line graph at the paragraph:

*Rising sea levels are eroding beaches in the South Pacific.*

The paragraph selected by our SIGHT system was regarded as relevant to the line




(a) A line graph showing a Change-Trend message




(b) A window in SIGHT system for line graph shows the textual summary for Figure 9.3a

Figure 9.3: The brief summary of a line graph in SIGHT system

graph by one of our two human evaluators. If the user then types Ctrl-Z, the intended message displayed in Figure 9.3a is recognized and conveyed to the user as spoken language by JAWS.

## 9.2 Summary

Our research has extended SIGHT so that it can handle simple line graphs as well as simple bar charts. Future work will address constructing a slightly longer summary that expands on the graphic's intended message by including salient features of the line graph and extending SIGHT's interaction facility to provide the user with further information about the graphic, both of which SIGHT does for bar charts.

# Chapter 10

# FUTURE WORK

We have discussed our methods for recognizing the intended messages for line graphs and how we can identify the paragraphs in a document that are most relevant to an information graphic. The SIGHT system has been enhanced to include both results so that a blind user can be prompted about the presence of a line graph at the most relevant place in a document and can be provided with an English language realization of its intended message. The following sections discuss our future work based on the intended message recognition and most relevant paragraph identification.

## 10.1 Follow-up questions for line graphs

Seniz Demir has extended Elzer's work[32, 31] on bar charts to produce longer natural language summaries that included salient features of the graphic[25, 24, 26]. In addition, she provided a mechanism for responding to requests for follow-up information[25, 24, 26].

However, extended summaries of line graphs will differ from summaries of bar charts since the features of line graphs are different. Thus research is needed to identify these features and generate summaries that take the most salient ones into account. Beyond the intended messages and an extended summary, the follow-up questions[1] may be

[1]The follow up question requests extra information about an information graphic besides the brief summary based on the intended messages. For example, the user might be interested in the volatility of a line graph and want to see a natural language sentence conveying this type of information.

different for line graphs and bar charts. A simple example is that a line graph might have a Contrast-Trend-Last-Segment intended message showing a most recent small segment change from the long trend segment preceding it. This intended message might motivate the viewer to raise a prediction question such as "How long will it take to return to the same previous level", and thus requires us to measure the slopes and lengths of the two segments, draw conclusions and produce a follow-up natural language response accordingly.

Pinker's work[71] discussed the four procedures for a reader to make predictions. They are: *MATCH* process that recognizes individual graphs as belonging to a particular type, a *message assembly* process that creates a conceptual message out of the instantiated graph schema, an *interrogation* process that retrieves or encodes new information on the basis of conceptual questions, and a set of *inferential processes* that apply mathematical and logical inference rules to the entries of the conceptual message. Although in our current research we are not doing prediction on line graphs, it would be possible to extend our work to incorporate the last two steps of Pinker's procedure and make predictions. Since line graphs are primarily used to show trends, predictions that extend the time series can be useful for either blind users or for question answering systems.

## 10.2 Multimodal document summarization

We showed in Chapter 8 that our research on identification of the most relevant paragraphs can provide good results so the brief summary of a line graph built from its intended message can be inserted at the most coherent place in an article. The multimodal document summarization techniques considering the more complete article can be investigated. The summarization can even take advantage of the extra information conveyed in the information graphics other than the intended message to augment the article, such as the volatility of the time series, the salience of a data point and the forecast.

## 10.3 Retrieval of line graphs with a mixture model

Our research on recognizing the intended message of a line graph and identifying the most relevant paragraphs can also be extended to information graphic retrieval task. Information graphics retrieval is a cross area between text retrieval and image retrieval. The goal of information graphics retrieval is to retrieve the appropriate information graphic according to its relevance to the user query. The query can be put into two categories as is the case for image retrieval: query by words and query by example. In query by words, the query is represented as a bag of keywords, which requires that the images be accompanied with metadata such as caption, keywords, or descriptions. Query by example can be regarded as given an information graphic, find all the information graphics containing the same subject or the same message. The future work discussed in this section doesn't facilitate query by example.

Information graphics are different from either pure text retrieval or pure image retrieval in two ways:

1. Information graphics contain both image information such as a line graphic or a bar chart and text information such as caption, descriptions or text-in-graphic. All parts contribute to the whole information graphic. The retrieval system should incorporate both the image and text components in the information graphics and consider them both in deciding whether to retrieve the graphic.

2. The intended message of a line graph captures the high-level communicative goal of the graphic and thus impacts when the information graphic should be retrieved in response to a user query.

3. Information graphics have a communicative goal that often supplements, and is supported by, the article's text. Thus the document is another component relevant to, but outside, the information graphic and should be considered.

Therefore there are three major components that we hypothesize potentially contribute to retrieval of information graphics: the text of the graphic (caption/description/text-in-graphic), the communicative goal of the graphic (intended message), and the article in which the information graphic appears. Figure 10.1 shows the probabilistic mixture model which can be potentially useful to combine the three major components together.

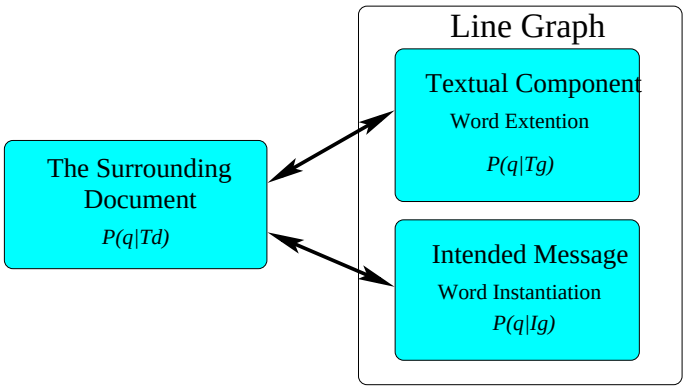



Figure 10.1: The relationship of the three components

When a query is given, we assume that the above three components all contribute to the relevance of an information graphic to the query. It could be that the intended message of the information graphic is reflected in the query, or part of the caption/description is covered in the query, or the article associated with the information graphic covers the keywords in the query. So here we define a probability term $p(G|q)$ representing the relevance of the information graphic to the query. And $p(G|q)$ is a combination of $p(I_G|q)$, $p(T_G|q)$ and $p(T_D|q)$, which are the relevance of the intended message of an information graphic, the relevance of the caption/description component of an information graphic, and the relevance of the article containing the information graphic, respectively, to a given query. We assume that $p(I_G|q)$,$p(T_G|q)$ and $p(T_D|q)$ have different weights $\alpha$,$\beta$,$\gamma$ respectively, indicating their importance. And we hypothesize that $p(G|q)$ should be close to all of the three components by minimizing its distance from them:

$$\alpha[p(G|q) - p(I_G|q)]^2 + \beta[p(G|q) - p(T_G|q)]^2 + \gamma[p(G|q) - p(T_D|q)]^2$$

Taking the derivative w.r.t $p(G|q)$ and setting it to 0, we obtain

$$p(G|q) = \frac{\alpha p(I_G|q) + \beta p(T_G|q) + \gamma p(T_D|q)}{\alpha + \beta + \gamma}$$

If we set $\alpha + \beta + \gamma = 1$, the denominator becomes 1 and $p(G|q)$ becomes

$$p(G|q) = \alpha p(I_G|q) + \beta p(T_G|q) + \gamma p(T_D|q)$$

After applying Bayes rule, we extend it further into

$$\begin{aligned} p(G|q) &= \alpha p(I_G|q) + \beta p(T_G|q) + \gamma p(T_D|q) \\ &= \alpha \frac{p(q|I_G)p(I_G)}{p(q)} + \beta \frac{p(q|T_G)p(T_G)}{p(q)} + \gamma \frac{p(q|T_D)p(T_D)}{p(q)} \\ &\propto \alpha p(q|I_G)p(I_G) + \beta p(q|T_G)p(T_G) + \gamma p(q|T_D)p(T_D) \end{aligned} \tag{10.1}$$

where $\alpha + \beta + \gamma = 1$.

The $p(q|I_G)$, $p(q|T_G)$ and $p(q|T_D)$ come from the different query likelihood models of the three components. They can be estimated by either a generative model or a boolean retrieval model. Within the generative model, an expansion word list for each component can be learned as we did in Chapter 8. The $p(I_G)$, $p(T_G)$ and $p(T_D)$ are the prior probabilities of the three components. $p(I_G)$ can be estimated by the proportion of graphics with messages in this message category in the corpus; $p(T_G)$ and $p(T_D)$ can be estimated simply as a uniform distribution, or can be estimated by their prior preference after some analysis. For example, $p(T_D)$ can be calculated according to the source of the information graphic, which means that the more authoritative news source has a higher prior preference[103].The $\alpha$, $\beta$ and $\gamma$ can be estimated by introducing a loss function between the $p(G|q)$ and the true rank assigned by human evaluators to graphics in response to a user query. After minimizing the difference between the estimated rank and the true rank, we could probably find a set of reasonable parameters.

This proposed mixture model can be further improved. Our research on Chapter 8 can help reduce the document component $T_D$ from the whole document into a set of relevant paragraphs which can thus remove the noise from the other topics in the document.

The generative probability $p(q|I_G)$ can be assigned by measuring which words have a strong relationship with which intended message category.

## 10.4 Summary

This chapter discussed potential future work which uses the two main research topics covered in this thesis: recognizing the intended message of a line graph and identifying the paragraphs that are most relevant to an information graphic. The future research can contribute to multiple areas including the SIGHT system, multimodal document summarization, and the retrieval of information graphics from a digital library.

# Chapter 11

# SUMMARY AND CONCLUSION

Information graphics are non-pictorial graphics such as bar charts and line graphs. They appear in popular media such as *New York Times*, *Businessweek*, *Wall Street Journal*, etc. as well as in scientific articles. Usually they are one part of a multimodal document which contains both the textual article and the information graphic. The majority of information graphics in popular media, such as newspapers and magazines, have a message that they are intended to convey. This dissertation discussed our method of recognizing the intended message of a line graph using Bayesian network, based on the communicative signals that appear in the graphic. The different categories of intended message are introduced in Chapter 3. The variety of communicative signals used by the graphic designers are discussed in Chapter 4. Our system first segments the line graph into visually distinguishable trends with a Graph Segmentation Module as discussed in Chapter 6. Then a Suggestion Generation Module which generates intended message candidates and the Bayesian network which does the probabilistic inference are invoked as discussed in Chapter 7. On a training set containing 240 line graphs collected from multiple sources and annotated by three human annotators, our system produced an overall 72.08% accuracy under leave-one-out cross validation. In the 173 correct instances, our system not only recognizes the correct intended message as agreed by the three human annotators but also identifies the correct parameters.

To apply our message recognition system to several application projects, we developed a method for identifying paragraphs that are relevant to an information graphic

in a document. The method uses the KL divergence similarity measure and an expansion word list learned from a set of training articles containing line graphs. This work was discussed in Chapter 8. On a test set containing 100 articles, our two methods P-KLE (KL divergence with expansion word list to augment the textual component) and P-KLEM (based on the P-KLE but using a linear combination of the scores from a paragraph and the best sentence in the paragraph as the final score to rank a paragraph) chose the most relevant paragraph as selected by either of the two human annotators with 57% and 64% success rate; and they chose a relevant paragraph as annotated by the two human annotators with 67% and 78% success rate. The $nDCG$ criteria also indicates that our two methods provide significant improvement over the baseline methods.

Our work on recognizing the intended message of a line graph and identifying the most relevant paragraphs for a line graph in a multimodal document have been embedded in the SIGHT system which provides a blind user with access to the information graphics. It can prompt the blind user about the presence of a line graph at the most relevant paragraph and can generate a brief textual summary of a line graph based on its recognized intended message. Besides the SIGHT system, our work can be further applied to a textual summarization system of multimodal documents, and retrieval of information graphics from a digital library.

# BIBLIOGRAPHY


[1] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2037–2040, 2002.

[2] R. J. Beckman and R. D. Cook. Testing for two-phase regressions. In *Technometrics*, pages 65–69, 1979.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st edition, 2007.

[4] David C. Bradley, Garry M. Steil, and Richard N. Bergman. OOPSEG: a data smoothing program for quantitation and isolation of random measurement error. In *Computer Methods and Programs in Biomedicine*, pages 67–77, 1995.

[5] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares Jr., and David Haussler. Knowledge based analysis of microarray gene expression data by using support vector machines. In *Proceedings of the National Academy of Sciences*, pages 262–267, 2000.

[6] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[7] Christopher J. C. Burges. Ranking as learning structured outputs. In *Proceedings of the NIPS 2005 workshop on Learning*, pages 7–11, 2005.

[8] Christopher J.C. Burges, Robert Ragno, and Quec Viet Le. Learning to rank with nonsmooth cost functions. In *NIPS 2006*, pages 193–200, 2006.

[9] Davide Buscaldi, Paolo Rosso, Jos Manuel Gmez Soriano, and Emilio Sanchis. Answering questions with an n-gram based passage retrieval engine. In *Journal of Intelligent Information Systems*, pages 113–134, 2010.

[10] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, 2008.

[11] Sandra Carberry, Stephanie Elzer, , and Seniz Demir. Information graphics: An untapped resource for digital libraries. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 581–588, 2006.

[12] C. Melody Carswell, Cathy Emery, and Andrea M. London. Stimulus complexity and information integration in the spontaneous interpretations of line graphs. In *Applied Cognitive Psychology*, pages 341–357, 1993.

[13] Gianfranco Cellarosi and Stefano Lodi. Detecting outbreaks by time series analysis. In *Proceedings of the 15th IEEE symposium on Computer-Based Medical Systems*, pages 159–164, 2002.

[14] Daniel Chester and Stephanie Elzer. Getting computers to see information graphics so users do not have to. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*, pages 660–668, 2005.

[15] Tak chung Fu, Fu lai Chung, Robert Luk, and Chak man Ng. Stock time series pattern matching: Template-based vs. rule-based approaches. In *Engineering Applications of Artificial Intelligence*, pages 347–364, 2007.

[16] Charles L. A. Clarke, Gordon V. Cormack, Thomas R. Lynam, and Egidio L. Terra. Question answering by passage selection. In *Advances in Open Domain Question Answering*, pages 259–283, 2008.

[17] Kevyn Collins-Thompson and Jamie Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 303–310, 2007.

[18] Marc Corio and Guy Lapalme. Generation of texts for information graphics. In *Proceedings of the 7th European Workshop on Natural Language Generation EWNLG'99*, pages 49–58, 1999.

[19] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 2009.

[20] Bruce W. Croft, Stephen Cronen-Townsend, and Victor Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)*, 2001.

[21] Hang Cui, Renxu Sun, Keya Li, Min yen Kan, and Tat seng Chua. Question answering using dependency relations. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 40–407, 2005.

[22] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 1st edition, 2009.

[23] Seniz Demir, Sandra Carberry, and Stephanie Elzer. Effectively realizing the inferred message of an information graphic. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing RANLP (2007)*, pages 150–156, 2007.

[24] Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. A discourse-aware graph-based content-selection framework. In *Proceedings of the 6th International Natural Language Generation Conference(INLG'10)*, pages 17–25, 2010.

[25] Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, and Kathleen F. McCoy. Interactive sight demo: textual summaries of simple bar charts. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (ASSETS 2010)*, pages 267–268, 2010.

[26] Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, and Kathleen F. McCoy. Interactive sight into information graphics. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, pages 16:1–16:10, 2010.

[27] Carlos A.R. Diniz and Luis Corte Brochi. Robustness of two-phase regression tests. In *REVSTAT - Statistical Journal*, pages 1–18, June 2005.

[28] Stephanie Elzer, Sandra Carberry, Daniel Chester, Seniz Demir, Nancy Green, Ingrid Zukerman, and Keith Trnka. Exploring and exploiting the limited utility of captions in recognizing intention in information graphics. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 223–230, 2005.

[29] Stephanie Elzer, Sandra Carberry, and Seniz Demir. Communicative signals as the key to the automated understanding of simple bar charts. In *Proceedings of the Fourth International Conference on the Theory and Application of Diagrams*, pages 25–39, 2006.

[30] Stephanie Elzer, Sandra Carberry, Ingrid Zukerman, Daniel Chester, Nancy Green, and Seniz Demir. A probabilistic framework for recognizing intention in information graphics. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1042–1047, 2005.

[31] Stephanie Elzer, Edward Schwartz, Sandra Carberry, Daniel Chester, Seniz Demir, and Peng Wu. A browser extension for providing visually impaired users access to the content of bar charts on the web. In *Proceedings of Third International Conference on Web Information Systems and Technology (WebIST)*, pages 59–67, 2007.

[32] Stephanie Elzer, Edward Schwartz, Sandra Carberry, Daniel Chester, Seniz Demir, and Peng Wu. Accessible bar charts for visually impaired users. In *Proceedings of the Fourth Annual IASTED International Conference on Telehealth and Assistive Technologies*, pages 55–60, 2008.

[33] Leo Ferres, Avi Parush, Zhihong Li, Yandu Oppacher, and Gitte Lindgaard. Representing and querying line graphs in natural language: The igraph system. In *Smart Graphics 6th International Symposium*, pages 248–253, 2006.

[34] Leo Ferres, Petro Verkhogliad, and Louis Boucher. (natural language) interaction with graphical representations of statistical data. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility*, pages 132–133, 2007.

[35] Leo Ferres, Petro Verkhogliad, Gitte Lindgaard, Louis Boucher, Antoine Chretien, and Martin Lachance. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 67–74, 2007.

[36] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Wai Lam. News sensitive stock trend prediction. In *Advances in Knowledge Discovery and Data Mining*, pages 481–493, 2002.

[37] Frederick J Gravetter and Frederick J Gravetter. *Essentials of Statistics for the Behavioral Sciences*. Wadsworth Publishing, 7th edition, 2010.

[38] Nancy L. Green, Giuseppe Carenini, Stephan Kerpedjiev, Joe Mattis, Johanna D. Moore, and Steven F. Roth. Autobrief: an experimental system for the automatic generation of briefings in integrated text and information graphics. *Int. J. Hum.-Comput. Stud.*, pages 32–70, 2004.

[39] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42, 1999.

[40] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. In *Machine Learning*, pages 389–422, 2002.

[41] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johnna Tikanmaki, and Hannu T.T. Toivonen. Time series segmentation for context recognition in mobile devices. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 203–210, 2001.

[42] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[43] Bernard Hugueney and Bernadette Bouchon-Meunier. Time-series segmentation and symbolic representation, from process-monitoring to data-mining. In *Computational Intelligence, Theory and Applications*, pages 118–123, 2001.

[44] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, pages 422–446, 2002.

[45] Irvin R. Katz, Xiaoming Xi, Hyun-Joo Kim, and Peter C-H. Cheng. Graph structure supports graph description. In *Proceedings of the Twenty-fourth Annual Meeting of the Cognitive Science Society*, pages 157–162, 2002.

[46] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proceedings of IEEE International Conference on Data Mining*, pages 289–296, 2001.

[47] Mahboob Alam Khalid and Suzan Verberne. Passage retrieval for question answering using sliding windows. In *Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 26–33, 2008.

[48] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[49] Stephen M. Kosslyn. *Elements of Graph Design*. W.H. Freeman and Company, 1993.

[50] A. Parush L. Ferres, S. Roberts, and G. Lindgaard. Helping people with visual impairments gain access to graphical information through natural language: The igraph system. In *Proceedings of the 10th International Conference on Computers for Handicapped Persons, Lecture Notes in Computer Science. Springer-Verlag*, pages 1122–1130, 2006.

[51] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, 2001.

[52] Fu lai Chung, Tak-Chung Fu, Vincent Ng, and Robert W. P. Luk. An evolutionary approach to pattern-based time series segmentation. In *IEEE Transactions on Evolutionary Computation*, pages 471–489, 2004.

[53] Adenike M. Lam-Adesina and Gareth J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–9, 2001.

[54] Victor Lavrenko. *A Generative Theory of Relevance*. Springer, 2008.

[55] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *proceedings of the 6th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, pages 37–44, 2000.

[56] Kyung Soon Lee, W. Bruce Croft, and James Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242, 2008.

[57] E. Levy, J. Zacks, B. Tversky, and D. Schiano. Gratuitous graphics? putting preferences in perspective. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 42–49, 1996.

[58] Xiaoyan Li and Zhigang Zhu. Enhancing relevance models with adaptive passage retrieval. In *Advances in Information Retrieval*, pages 463–471, 2008.

[59] Xiaoyan Liu, Zhenjiang Lin, and Huaiqing Wang. Novel online methods for time series segmentation. In *IEEE Transactions on Knowledge and Data Engineering*, pages 1616–1626, December 2008.

[60] Xiaoyong Liu and W. Bruce Croft. Passage retrieval based on language models. In *Proceedings of the eleventh international conference on Information and knowledge management CIKM '02*, pages 375–382, 2002.

[61] Wen-Hsiang Lu, Chia-Ming Tung, and Chi-Wei Lin. Question intention analysis and entropy-based paragraph extraction for medical question answering. In *6th world congress of biomechanics*, pages 1582–1586, 2010.

[62] Yuanhua Lv and Chengxiang Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 299–306, 2009.

[63] Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–618, 2003.

[64] Ulla Maichle. Cognitive processes in understanding line graphs. In *Comprehension of Graphics*, pages 207–226. North-Holland/Elsevier Science, 1994.

[65] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 2nd edition, 2008.

[66] Vibhu O. Mittal, Giuseppe Carenini, Johanna D. Moore, and Steven Roth. Describing complex charts in natural language: a caption generation system. *Computational Linguistics - Special issue on natural language generation*, pages 431–467, 1998.

[67] Vibhu O. Mittal, Steven Rothy, Johanna D. Moore, Joe Mattisy, and Giuseppe Carenini. Generating explanatory captions for information graphics. In *Proceedings of the 14th international joint conference on Artificial intelligence*, 1995.

[68] Max Moldovan. Testing for a two-phase multiple regression. In *Transport and Telecommunication*, pages 35–42, 2003.

[69] Norsys Software Corp.: Netica, 2005.

[70] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009.

[71] Steven Pinker. A theory of graph comprehension. In *Artificial Intelligence and the future of testing*, pages 73–126, 1990.

[72] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, pages 185–208, 1999.

[73] Franois Portet, Ehud Reiter, Jim Hunter, and Somayajulu Sripada. Automatic generation of textual summaries from neonatal intensive care data. In *Artificial Intelligence in Medicine*, pages 227–236, 2007.

[74] Richard E. Quandt. The estimation of the parameters of a linear regression system obeying two separate regimes. In *Journal of the American Statistical Association*, pages 873–880. American Statistical Association, 1958.

[75] Richard E Quandt. Tests of the hypothesis that a linear regression system obeys two separate regimes. In *Journal of the American Statistical Association*, pages 324–330. American Statistical Association, 1960.

[76] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1st edition, 1992.

[77] Ehud Reiter. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 97–104, 2007.

[78] Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. Choosing words in computer-generated weather forecasts. In *Artificial Intelligence*, pages 137–169, 2005.

[79] Lucia Sacchi, Cristiana Larizza, Carlo Combi, and Riccardo Bellazzi. Data mining with temporal abstractions: learning rules from time series. In *Data Mining and Knowledge Discovery*, pages 217–247, 2007.

[80] Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 576–584, 2004.

[81] Wolfgang Schnotz and R. W. Kulhavy. *Comprehension of Graphics*. North Holland, 1994.

[82] Priti Shah and James Hoeffner. Review of graph comprehension research: Implications for instruction. In *Educational Psychology Review*, pages 47–69, 2002.

[83] Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. Generating english summaries of time series data using the gricean maxims. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196, 2003.

[84] Renxu Sun, Chai-Huat Ong, and Tat-Seng Chua. Mining dependency relations for query expansion in passage retrieval. In *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 382–389, 2006.

[85] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[86] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 162–169, 2006.

[87] Wei-Guang Teng, Ming-Syan Chen, and Philip S. Yu. A regression-based temporal pattern mining scheme for data streams. In *Proceedings of the 29th VLDB Conference*, pages 93–104, 2003.

[88] Evimaria Terzi and Panayiotis Tsaparas. Efficient algorithms for sequence segmentation. In *SIAM International Conference on Data Mining*, pages 314–325, 2006.

[89] G.L. Tietjen, R.H. Moore, and R.J. Beckman. Testing for a single outlier in simple linear regression. In *Technometrics*, pages 717–721, 1973.

[90] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley, 3rd edition, 2010.

[91] Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. Evaluating paragraph retrieval for why-qa. In *Advances in Information Retrieval*, pages 669–673, 2008.

[92] Elisabeth Vieth. Fitting piecewise linear regression functions to biological responses. In *Journal of Applied Physiology*, pages 390–396, 1989.

[93] Dominic Widdows. Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 136–143.

[94] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[95] Kenji Yamanishi and Jun ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 676–681, 2002.

[96] Rong Yan, Alexander Hauptmann, and Rong Jin. Multimedia search with pseudo-relevance feedback. In *Image and Video Retrieval*, pages 649–654, 2003.

[97] Rong Yan, Alexander G. Hauptmann, and Rong Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 343–346, 2003.

[98] Hong Yu and Minsuk Lee. Accessing bioscience images from abstract sentences. *Bioinformatics*, 22(14):547–556, 2006.

[99] Jin Yu, Jim Hunter, Ehud Reiter, and Somayajulu Sripada. Recognizing visual patterns to communicate gas turbine time-series data. In *Proceedings of 22nd SCAI International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (ES2002)*, pages 105–118, 2002.

[100] Jin Yu, Ehud Reiter, Jim Hunter, and Chris Mellish. Choosing the content of textual summaries of large time-series data sets. In *Natural Language Engineering*, pages 25–49, 2007.

[101] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th international conference on World Wide Web*, pages 11–18, 2003.

[102] Jeff Zacks and Barbara Tversky. Bars and lines: a study of graphic communication. In *Memory and Cognition*, pages 1073–1079, 1999.

[103] Chengxiang Zhai. *Statistical Language Models for Information Retrieval*. Morgan and Claypool Publishers, December 2008.

[104] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 403–410, 2001.

[105] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.

[106] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, pages 179–214, 2004.

[107] Wei Zhou, Clement yu, Neil Smalheiser, Vetle Torvik, and Jie Hong. Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. In *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 655–662, 2007.

# Appendix A

# RESAMPLING

The conversion of the line graph from a GIF file to an XML representation is done by the Visual Extraction Module[14]. The Visual Extraction Module captures the coordinates of the pixels at the two endpoints of any straight segment in the line graph and may cluster a set of sample points around some location, as shown in the solid circles in the top graph in Figure A.6 and the dashed circles in the top graph in Figure A.7. It is because the VEM tries to capture all straight lines in the original image file; if at some location there are multiple pixels with different slopes, the VEM will capture all of them even if they are very close to one another (by one or two pixels). While this set of sample points fully describes the line graph, it does not satisfy the needs of the Graph Segmentation Module which applies a set of statistical tests on the series of sampling points. Some of the statistical tests such as the *F test*, *Q test* and *Runs test* require us to apply them on a data sampling that is close to a uniform distribution on the x axis. Thus to implement our methodology for recognizing a line graph's intended message in the SIGHT system, we need to perform a resampling procedure on the result provided by the VEM. The resampling procedure tries to achieve the following goals:

1. The distribution of the resampled points on the x axis should be close to a uniform distribution. At a minimum, there shouldn't be a large proportion of data points clustered together, since such a distribution will influence the results of the statistical tests in the Graph Segmentation Module.

2. The resampled points should have little data loss from the original line graph.

3. We will refer to a local peak or valley such as the circled points in Figure A.1 as outstanding points. The resampling should keep the outstanding points as much as possible, so that those data points can be used as potential splitting points by the Graph Segmentation Module.

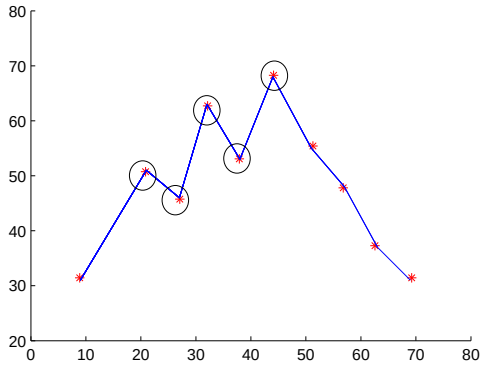



Figure A.1: The circled points are regarded as outstanding points which may be used as splitting points by Graph Segmentation Module

## A.1 Sampling with the same interval

A straightforward sampling method to achieve the first goal is to specify the total number of data points we need, and then do the sampling with the same interval. So if we describe the line graph as $f(t)$ where $0 \leq t \leq T$, the sampling will provide a pre-specified $k+1$ sampling points (including the starting and ending points which are $f(0)$ and $f(T)$ respectively) where $t_{i+1} - t_i = T/k$; although this sampling method provides a uniformly spaced series of new sampled points, it tends to ignore the shape of the line graph so that the outstanding points will often be missed. Thus a different method is needed.

## A.2 ARIMA/GARCH sampling

To capture the outstanding points, we need to detect outstanding points when we perform the resampling; to make the sampling be uniform, we need to try to do the sampling at pre-specified intervals. These two goals may conflict with each other. Thus we

designed a new sampling strategy based on the ARIMA/GARCH model that attempts to balance these criteria.

The ARIMA (autoregressive integrated moving average) and GARCH (generalized autoregressive conditional heteroskedasticity)[90] are widely used in econometrics for time series analysis. They are basically linear regression models that use the lag $l$ data to model the current data point, where "lag $l$" refers to the $l$ data points immediately preceding the one being modelled. ARIMA uses the old data points to model the mean of the difference between the current data point and the preceding data point, and GARCH uses them to model the variance.

ARIMA is based on the ARMA (autoregressive moving average) model. ARMA and GARCH both assume that the data series is stationary, where stationarity is defined as both the mean of data point $d_t$ at location $t$ and the covariance between $d_t$ and $d_{t-l}$ are time invariant, where $l$ is an arbitrary integer. Normally this condition is not satisfied by a given data series. However, the differences in the data series, defined as $r_t = d_t - d_{t-1}$, can satisfy the stationarity condition. The ARMA model applied on the data series represented by the differences $r_t$ is then named ARIMA.

The general $ARMA(p, q)$ model is of the form

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + a_t - \sum_{j=1}^{q} \theta_j a_{t-j} \tag{A.1}$$

where $r_t$ is the difference between the true value of the previous data point and the true value of the point at time $t$, $r_{t-i}$ are the differences in the true values of the previous consecutive points, $\{a_t\}$ is a white noise series, and $p$ and $q$ are nonnegative integers. The $ARMA(p, q)$ model is trying to model $r_t$ as a function of the preceding $r_{t-i}$ and calculated noise $a_{t-j}$ which is the difference between the true value of $r_t$ and its predicted value.

The $GARCH(m, s)$ model assumes that the actual value of $r_t$ is normally distributed around the predicted value of $r_t$ with standard deviation $\sigma_t$. To compute $\sigma_t$, the

GARCH model uses the predicted standard deviation $\sigma_{t-j}$ for the preceding points and the difference $b_{t-i}$ between the actual and predicted values of $r_{t-i}$.

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i b_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 \tag{A.2}$$

After choosing $m$, $s$, $p$ and $q$, the coefficients $\phi_i$, $\theta_j$, $\alpha_i$ and $\beta_j$ of $ARMA(p,q)$ and $GARCH(m,s)$ can be calculated by maximum likelihood estimation. But the open question is how to choose the $p, q$ in ARMA and $m, s$ in GARCH[1]. For this, we refer to the ACF (autocorrelation function) and the PACF (partial autocorrelation function). Given a data series $n_1, ..., n_T$, ACF is the correlation coefficient between the data series $n_{l+1}, ..., n_T$ and the lag $l$ data series $n_1, ..., n_{T-l}$ where $n_T$ is the last element of the series. It is of the form

$$ACF_l = \frac{Cov(n_t, n_{t-l})}{\sqrt{Var(n_t)Var(n_{t-l})}}$$

where $-1 \leq ACF_l \leq 1$ and $ACF_0 = 1$. If $ACF_l$ is equal to $0$, it means there is no correlation between the two data series. A statistical test is applied to determine if the $ACF_l$ is significantly different from $0$.

The partial autocorrelation function PACF considers multiple AR (autoregressive) models in consecutive order:

$$AR(1): \; n_t = \phi_{0,1} + \phi_{1,1} n_{t-1} + \epsilon_{1,t}$$

$$AR(2): \; n_t = \phi_{0,2} + \phi_{1,2} n_{t-1} + \phi_{2,2} n_{t-2} + \epsilon_{2,t}$$

$$AR(3): \; n_t = \phi_{0,3} + \phi_{1,3} n_{t-1} + \phi_{2,3} n_{t-2} + \phi_{3,3} n_{t-3} + \epsilon_{3,t}$$

$$AR(4): \; n_t = \phi_{0,4} + \phi_{1,4} n_{t-1} + \phi_{2,4} n_{t-2} + \phi_{3,4} n_{t-3} + \phi_{4,4} n_{t-4} + \epsilon_{4,t}$$

where $n_t$ is the value of an element in the data series, the $n_{t-i}$ are the values of earlier elements in the series, the $\epsilon_{i,t}$ are the error terms, and the $\phi_{i,j}$ are the constants produced

[1]Note that $m$ and $s$ may be different, and $p$ and $q$ may be different, since they do not need to use the same number of lag points.

by the regression model. The models are of the form of a linear regression and can be estimated by the least-squares method. The computed value of $\phi_{p,p}$ is called the lag-p sample PACF of $n_t$, denoted as $PACF_p$. It shows the added contribution of $n_{t-p}$ to $n_t$ in the $AR(p)$ model over the $AR(p-1)$ model. Ideally we want an $AR(p)$ model where the lag-p sample PACF $\phi_{p,p}$ is not zero, but $\phi_{j,j}$ is close to zero for all $j > p$. A statistical test can be applied to determine whether adding an extra term $\phi_{p+1,p+1}n_{t-(p+1)}$ will produce a significantly better model (i.e. whether $\phi_{p+1,p+1}$ is significantly different from $0$).

To select the parameters $p$ and $q$ respectively in $ARMA(p,q)$ (see Equation A.1), we compute the $PACF_l$ and $ACF_l$ series for $r_t$; to select $m$ and $s$ respectively in $GARCH(m,s)$, we compute the $PACF_l$ and $ACF_l$ series for $(r_t - \hat{r_t})^2$ where $\hat{r_t}$ is the predicted value of $r_t$ from Equation A.1. Each of the four parameters are set to the lag number $k$ whose corresponding series are significant from $1$ to $k$ and insignificant for $k+1$. For example, if we compute $\phi_{1,1}$, $\phi_{2,2}$, $\phi_{3,3}$,...,$\phi_{k,k}$ for the $r_t$ data series and they are all significant but $\phi_{k+1,k+1}$ is not significant, then we choose $k$ as the value for $p$.

By applying ARIMA and GARCH to the $r_t$ data series of differences, we can model its estimated mean and standard deviation. Assuming a normal distribution, we can compute the probability of the real data point given the predicted mean and standard deviation produced by ARIMA and GARCH. For example, Figure A.3 shows a prediction window of size $6$ for a piece of the line graph in Figure A.2. The three solid lines shows the predicted mean of the data points and their one standard deviation upperbound/lowerbound. The dashed line shows the real location of the data points in the window. This example assumes that we have already sampled a data point at location $0$ and are trying to select the next sample point. In this example, the actual data point at location $2$ is higher than the others and could be an outstanding point, but location $3$ is where we want the sampled point to be because the ideal interval is $3$.

Since we want to balance the criteria of having a uniformly distributed sample with the criteria of having all of the outstanding points in the sample, we compute the product

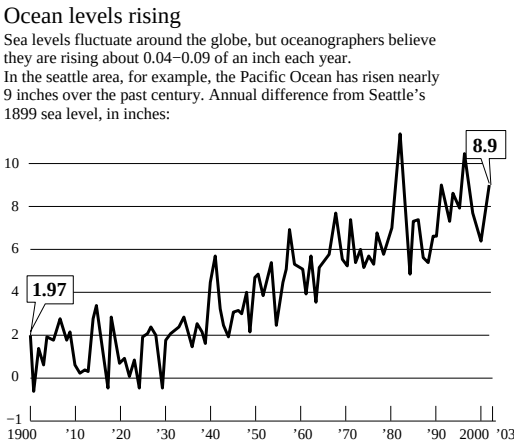



Figure A.2: A line graph which is sampled by the ARIMA/GARCH model

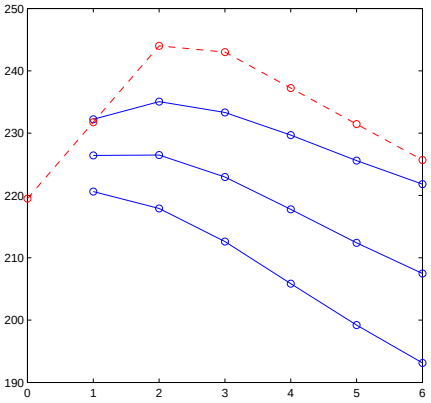



Figure A.3: Example showing the result of the ARIMA/GARCH for a piece of the line graph in Figure A.2

of two terms. The first term measures the preference for a location close to the center of the sampling window. It is measured as $f(x)$ where $x$ is normally distributed, centered at the middle of the window and with a standard deviation that is the ideal sampling interval.[2]

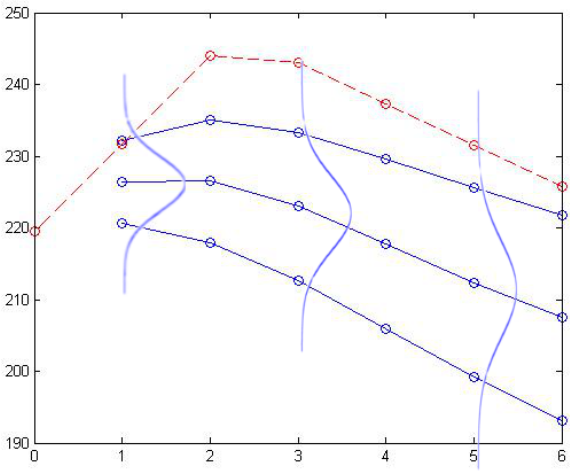
$$x \sim N(s, s^2) \text{ where } p(t) = \cdot \frac{1}{\sqrt{2\pi s^2}} e^{\frac{-(x-s)^2}{2s^2}} \tag{A.3}$$

The second term measures the preference for an outstanding data point and is measured as the cumulative probability of generating a data point between the predicted data point and the true value at that point according to the predicted mean and standard deviation, as shown in Figure A.4. This can be represented as a Gaussian error function $erf(r_t)$.

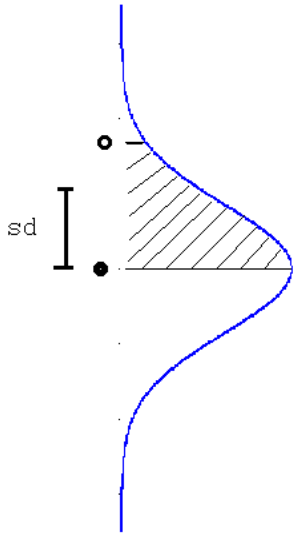
$$erf(z_x) = 2 \int_0^{|z_x - \hat{z_x}|} \frac{1}{\sqrt{2\pi\hat{\sigma}_x^2}} e^{\frac{-(z_x - \hat{z_x})^2}{2\hat{\sigma}_z^2}} \tag{A.4}$$

Given that the length of the line graph is $T$ and the pre-specified number of desired data points is $n$, the ideal interval of sampling is $T/n$, which is used as half of the window size, denoted as $s$. The reason that we use $T/n$ as half of the window size is because the

[2]We arbitrarily chose the standard deviation to be the same as the ideal sampling interval.




(a) The normal distribution for a data point given its predicted mean and standard deviation calculated from the ARIMA/GARCH model




(b) The calculation of Gaussian error function is double the cumulative density between a point and the mean of its normal distribution

Figure A.4: This example illustrates the Gaussian error function which is used as our measurement of whether a point is outstanding

ideal sampling location with interval $T/n$ is in the middle of the window, so that we can sample around this point within our window. The score of each potential sampling location $x$ on the x axis is calculated as

$$score(x) = p(x) \cdot erf(z_x) \tag{A.5}$$

This method covers the sampling strategy using the uniform interval which was discussed in Section A.1 as an extreme case. In our experiment, if we assign the standard deviation in Equation A.3 to a very small value, then $p(x)$ will have values close to $0$ at all locations except at $x = s$. Therefore according to Equation A.5, the first term dominates and the ARIMA/GARCH sampling strategy will always sample the data point at the ideal interval and thus reduces to the uniform sampling strategy.

| Location | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $erf(z_x)$ | 0.6415 | 0.9590 | 0.9470 | 0.8977 | 0.8515 | 0.7959 |
| $p(x)$ | 0.106 | 0.125 | 0.132 | 0.125 | 0.106 | 0.081 |
| score(x) | 0.0680 | 0.1199 | 0.1250 | 0.1122 | 0.0903 | 0.0645 |

Table A.1: The score of each location in Figure A.3

The scores for the data shown in Figure A.3 are shown in Table A.1. The top row records $erf(r_t)$ which measures whether the data point at each potential sampling location is outstanding. It is calculated according to the predicted mean and variance. The second row shows $p(x)$ which is the preference for that sampling location; it is a normal distribution centered at $s = 3$ and with standard deviation of $s = 3$. $score(x)$ is shown as the third row in this table. It indicates that even though the data point at location $2$ is more outstanding, our sampling scheme prefers the data point at $3$ because it is a balance between selecting outstanding points and the uniform sampling interval.

### A.3 Procedure of resampling in SIGHT

To resample the data provided by the VEM, the resampling system first interpolates between the consecutive sampling points given by the VEM to recover all of the pixel points in the whole line graph without regards to any original sampling points. The interpolation is applied on each pixel on the x axis of the original line graph and then generates a length $T$ interpolated time series where $T$ depends on the horizontal length of the original line graph in pixels. ARIMA and GARCH are implemented in MATLAB. The recovered data points are given to MATLAB which computes the parameters needed for ARIMA and GARCH as discussed in the previous section. Once the coefficients have been identified, MATLAB provides a prediction routine which gives us $r_t$ (the predicted difference from the previous data point) and from which the next data point can be predicted. Given the pre-specified desired number of data points $n$, the resampling process first calculates the window size which is double the size of the ideal interval $T/n$. Then the process moves from the left to right to sample points. It puts a window to the right of

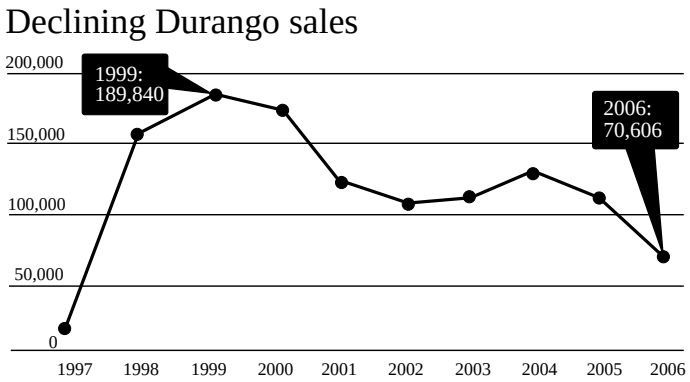



Figure A.5: A line graph which is sampled by the ARIMA/GARCH model

the most recent sampled data point, calculates the score of each potential sample location in the window, and scores them based on the preference for each location in the window and the measurement of whether the actual data point at this location is outstanding. The location in the window with the maximum score is chosen as the next sampling location and then the window moves to the right of the new location and the sampling run starts again with the new data points in the window. It continues until reaching the end of the line graph. The last point of the whole line graph must be chosen as well as the first point of the line graph. This sampling strategy gives us a result which balances capturing all the outstanding data points against the desire for a uniform sampling interval.

This sampling strategy may not generate exactly the same number of sampling points as pre-specified because it might not always sample with the same interval. If the line graph is jagged and has many outstanding points, it may sample more points than pre-specified to try to reduce the loss of data. In our system, we used the number of sample points provided by VEM as the number of desired sample points.

## A.4 Demonstration and analysis

We can see the advantage of the ARIMA/GRACH sampling strategy over the uniform sampling strategy from two examples. First, Figure A.6 shows the sampling results for the line graph in Figure A.5. The top one is the original sampling generated by the

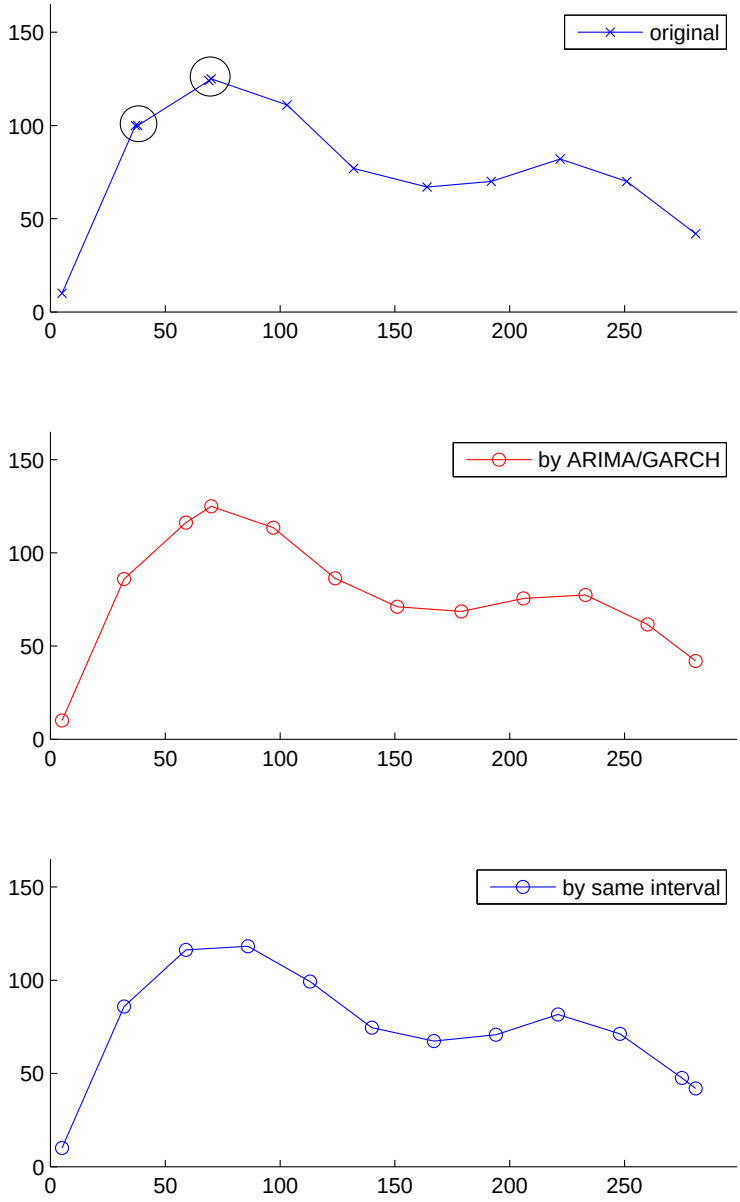



Figure A.6: The result provided by the VEM and the two resampling methods on Figure A.5

VEM. There are multiple sampling points around the circled locations. It is because the VEM tries to capture all straight lines in the original image file; if at some location, there are multiple pixels with different slopes, the VEM will capture all of them even if they are very close to one another (by one or two pixels). The second line graph shows the sampling from our ARIMA/GARCH resampling strategy. We can see that there is no location with multiple sample points clustered around it and the maximum point in the original line graph has been captured, which is eventually used as the splitting point in the Graph Segmentation Module. The third sampling is achieved by sampling with a uniform interval; we can see that the maximum point is not captured.

Figure A.7 shows the sampling results for a portion of the graph in Figure A.2. The top figure is the sampling generated by the VEM. We can see that there are many clustered data points as in the dashed circle. The bottom figure is the sampling result generated by a uniform sampling interval; although it managed to capture the maximum data point, it failed to capture other outstanding points that occur in the circled area. The plot in the middle is the sampling generated by the ARIMA/GARCH sampling strategy. It keeps the high points and the low points which are not captured by the uniform sampling strategies, and there is no clustering of data points.

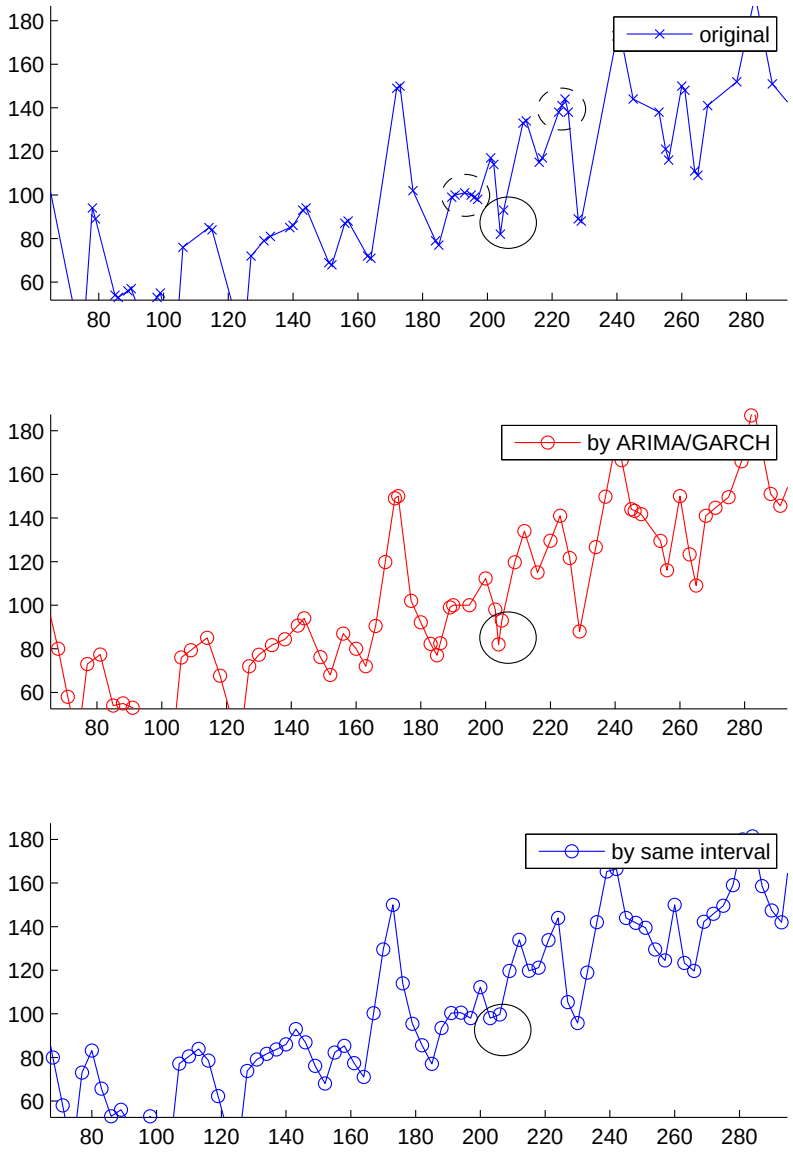



Figure A.7: The result provided by the VEM and the two resampling methods on a portion of Figure A.2

# Appendix B

# PERMISSION LETTERS

HUMAN SUBJECTS PROTOCOL
University of Delaware

Protocol Title: Important Content of Information Graphics

Principal Investigator
Name: Sandra Carberry
Contact Phone Number: 302-831-1954
Email Address: carberry@cis.udel.edu

Advisor (if student PI):
Name:
Contact Phone Number:
Email Address:

Other Investigators: Peng Wu (graduate student)

Type of Review:
Exempt ✓Expedited Full board

Exemption Category: 1 2 3 4 5 6

Minimal Risk: ✓ yes _____ no

Submission Date: June 10, 2009

| HSRB Approval Signature | Approval Date |
|---|---|
| Elizabeth Dugan Peloso | 6/12/09 |
| HS Number | Approval Next Expires |
| 09-694 | 6/11/10 |

Investigator Assurance:
By submitting this protocol, I acknowledge that this project will be conducted in strict accordance with the procedures described. I will not make any modifications to this protocol without prior approval by the HSRB. Should any unanticipated problems involving risk to subjects, including breaches of guaranteed confidentiality occur during this project, I will report such events to the Chair, Human Subjects Review Board immediately.

Signature of Investigator: _____

Date: _____

1. **Is this project externally funded? Yes**

If so, please list the funding source:NSF

2. **Project Staff**
Please list personnel, including students, who will be working with human subjects on this protocol (insert additional rows as needed):

| NAME | ROLE | HS TRAINING COMPLETE? |
|---|---|---|
| Mary Sandra Carberry | PI | Yes |
| Peng Wu | Graduate research assistant | In progress |
| | | |
| | | |
| | | |
| | | |
| | | |

3. **Special Populations**
Does this project involve any of the following:

Research on Children? No

Research with Prisoners? No

Research with any other vulnerable population (please describe)? No

4. **RESEARCH ABSTRACT** Please provide a brief description in LAY language (understandable to an 8th grade student) of the aims of this project.

The amount of information contained in digital libraries is growing rapidly, and research is producing sophisticated techniques for summarizing articles and for answering questions based on the content of articles in the library. Unfortunately, these techniques are limited to the textual content of articles; they cannot take a multimodal document's graphics into account when producing a summary and cannot utilize the graphics when answering questions. Our project has the goal of producing richer document summaries and better querying techniques. We are focusing on information graphics, such as bar charts and line graphs. When information graphics appear in popular media, they generally have a message that they are intended to convey. We are developing a computer system to recognize the intended message of an information graphic. This message will then be used along with an article's text when summarizing the article, and the message will also be used as a resource when answering questions based on the article.

We have already implemented a system for identifying the message conveyed by a simple bar chart. Our research is now addressing simple line graphs. In identifying the high-level message conveyed by a line graph, we must treat it as a sequence of visually distinguishable trends (as opposed to a large number of very short line segments). We have developed a system for dividing a line graph into such visually distinguishable trends (which we refer to as its segmentation). The purpose of the proposed study is to evaluate the results produced by our system.

5. **PROCEDURES** Describe all procedures involving human subjects for this protocol. Include copies of all surveys and research measures.

There are two separate parts to our proposed research study; the combination of the two parts will enable us to determine how successful our system is at producing a good segmentation of a line graph. In Part 1 of the study, Evaluating the Segmentation, subjects will be given a set of line graphs along with a candidate segmentation, some of which are produced by our system; the subjects will be asked to rate the quality of the segmentation on a 5-point scale. The subjects will not be judged in any way regarding the speed with which they respond or the kinds of responses that they provide. We are just interested in their rating of our system's results.

In Part 2 of the study, Comparing Segmentations, subjects will be given a set of line graphs and two candidate segmentations (one produced by our system and one produced by another method) and asked to choose the better segmentation. Once again, the subjects will not be judged in any way regarding the speed with which they respond or the kinds of responses that they provide. We only want to determine how our system's results compare with those produced by other methodologies.

6. **STUDY POPULATION AND RECRUITMENT**

Describe who and how many subjects will be invited to participate. Include age, gender and other pertinent information. Attach all recruitment fliers, letters, or other recruitment materials to be used.

The subjects will all be adults who will not be hurt in any way by their reluctance or refusal to participate. We will invite undergraduate students, graduate students, family, and friends to participate. In recruiting undergraduate students, we will contact students whom we know and announce our recruitment in undergraduate classes.

We anticipate needing at least 5 participants for each part of the study in order to obtain a sufficient set of data, but we will gladly accept more participants, up to as many as 20 participants for each part of the study.

Describe what exclusionary criteria, if any will be applied.

Subjects must be native English speakers or fluent in English. We have this requirement to insure that the subjects understand the task instructions.

Describe what (if any) conditions will result in PI termination of subject participation.

None

7. **RISKS AND BENEFITS**

Describe the risks to participants (risks listed here should be included in the consent document). If risk is more than minimal, please justify.

No risks to the subjects are foreseen.

What steps will be taken to minimize risks?

There are no risks to the subjects.

Describe any direct benefits to participants.

In limited cases, there may be some course extra credit offered for participation in the research study. Under no circumstances will a course grade be penalized as a result of a student's decision not to participate. Alternative extra credit will be offered to those students who do not wish to participate in the study. (The alternative extra credit will have the same weight as the study participation and require approximately the same amount of time from the subject.) Aside from the

extra credit possibility, there is no tangible benefit or reward for participation in the research study.

Describe any future benefits to this class of participants.
There are no future benefits to this class of participants.

If there is a Data Monitoring Committee (DMC) in place for this project, please describe when and how often it meets.
No data monitoring committee exists.

8. **COMPENSATION**
Will participants be compensated for participation?
Other than the extra credit possibility described above, participants will not be compensated for participation.

If so, please include details.
In limited cases, there may be some course extra credit offered for participation in the research study. Under no circumstances will a course grade be penalized as a result of a student's decision not to participate. Alternative extra credit will be offered to those students who do not wish to participate in the study. (The alternative extra credit will have the same weight as the study participation and require approximately the same amount of time from the subject.) Aside from the extra credit possibility, there is no tangible benefit or reward for participation in the research study.

9. **DATA**
Will subjects be anonymous to the researcher?
No.

If subjects are identifiable, will their identities be kept confidential?
Yes, identity of subjects will be kept confidential.

How and how long will data be stored?
The data will not be associated with identifying information about the subject who provided the data. Thus the researchers working on the project will be able to use copies of the data in their research. The consent forms will be stored in the office of Dr. Sandra Carberry and will be kept on file for three years from the date of the experiment.

How will data be destroyed?
We do not anticipate any need to destroy the collected data; note that all collected data will be anonymous with regard to which subject the data was obtained from.

How will data be analyzed and reported?
The data will be analyzed to determine the overall rating of the system's segmentations (Part I) and to determine whether the system's segmentation is rated better on average than the segmentation produced by the other methodology (Part 2).

10. **CONFIDENTIALITY**
Will participants be audiotaped, photographed or videotaped during this study?
No

How will subject identity be protected?
The collected data will not include any indication of the subject who provided the data. Consent

forms will be stored separately from the collected data; the consent forms will be stored in the office of Dr. Sandra Carberry.

Is there a Certificate of Confidentiality in place for this project? (If so, please provide a copy).
No

11. **CONSENT and ASSENT**

__X__ Consent forms will be used and are attached for review.

_____ Additionally, child assent forms will be used and are attached.

_____ Consent forms will not be used (Justify request for waiver).

12. **Other IRB Approval**

Has this protocol been submitted to any other IRBs?
No

If so, please list along with protocol title, number, and expiration date.

**Please submit this form to the e-mail address: hsrb-research@udel.edu**

# Consent Form

## Graph Segmentation Research Study
## Part 1: Evaluating the Segmentation

1. PURPOSE/DESCRIPTION OF THE RESEARCH

We are developing a computer system that can identify the intended message of an information graphic, such as a bar chart or a line graph. This message can be used along with the text of an article to produce a summary or to answer queries based on the article's content. We have developed a system for segmenting a line graph into a sequence of visually distinguishable trends, which is the first step in identifying its intended message. The purpose of this study is to evaluate the quality of the segmentations produced by our system. Participants in this study must be at least 18 years of age. We anticipate having between 5 and 20 participants in Part 1 of the study. There are no set time limits for completion of Part 1 of the study, but it is expected that it should take 1-2 hours.

**Study Procedures:**

You will be given a set of line graphs along with a candidate segmentation, some of which were produced by our system. In each case, you will be asked to rate the segmentation on a 5 point scale. You will not be evaluated in any way; we are not interested in how long it takes you to rate the segmentations. We are only interested in compiling the ratings of the segmentations for each line graph and averaging them to obtain an evaluation of our computer system.

2. CONDITIONS OF SUBJECT PARTICIPATION

We will not reveal your name in connection with any of our data. Data will be stored without any indication of which subject provided the data.

You may withdraw from the study at any time during your participation; if you wish us to destroy the data collected thus far from you, we will do so. If you decide to withdraw from the study, there will not be any penalties or adverse consequences. We will notify you of any significant findings that might affect your willingness to participate in this study.

3. RISKS AND BENEFITS

There are no potential risks to you if you participate in this experiment. If you refuse to participate, you will not be penalized. In most cases, there are no direct benefits to you if you participate in this experiment. In some cases, your course instructor may have offered you some extra course credit for your participation. If this is the case, your instructor will have clearly outlined the available amount of extra credit and the guidelines for your participation. If you do not wish to participate in the study, an alternative extra credit assignment will be provided. Under no circumstances will a course grade be penalized for your decision not to participate.





_____(Please initial after reading)

If at any point during the administration of the experiment, you change your mind and wish to withdraw, you may do so and the data collected so far will be discarded. If you have received course extra credit for your participation, that extra credit will not be revoked.

4. CONTACTS

If you have any questions, or wish to withdraw from the study, please contact Dr. Sandra Carberry by calling (302) 831-1954, by sending mail to carberry@cis.udel.edu, or by writing to her at the Department of Computer and Information Sciences, 103 Smith Hall, University of Delaware, Newark, DE 19716.

If you have any questions or concerns regarding the rights of individuals who agree to participate in research, please contact the Chair of the University of Delaware Human Subjects Review Board at 302-831-2136.

5. SUBJECT'S ASSURANCES

Participation in this study is voluntary. It is your decision whether you want to participate. If you refuse to participate, you will not be penalized. If at any point during the experiment, you change your mind and wish to withdraw, you may do so. If you request to have the data collected from you destroyed, we will do so.

Experiment results and consent forms will be stored in Dr. Sandra Carberry's office to ensure confidentiality. Data will be stored for at least three years. A copy of this consent form will be provided for you.

6. CONSENT SIGNATURES

Please sign below if you agree to participate in this study:

_____ Date: _____

(Participant signature)

_____

(Please print your name)

UNIVERSITY OF DELAWARE
APPROVED BY HSRB

6/12/09 - 6/11/10 EOP

DATE



_____(Please initial after reading)

# Consent Form

Graph Segmentation Research Study
Part 2: Comparing Segmentations

## 1. PURPOSE/DESCRIPTION OF THE RESEARCH

We are developing a computer system that can identify the intended message of an information graphic, such as a bar chart or a line graph. This message can be used along with the text of an article to produce a summary or to answer queries based on the article's content. We have developed a system for segmenting a line graph into a sequence of visually distinguishable trends, which is the first step in identifying its intended message. The purpose of this study is to evaluate the quality of the segmentations produced by our system. Participants in this study must be at least 18 years of age. We anticipate having between 5 and 20 participants in Part 2 of the study. There are no set time limits for completion of Part 2 of the study, but it is expected that it should take no more than 1 hour.

**Study Procedures:**

You will be given a set of line graphs along with two candidate segmentations for each. For each line graph, you will be asked to choose the segmentation that you feel is a better representation of visually distinguishable trends in the graphic. You will not be evaluated in any way; we are not interested in how long it takes you to choose a segmentation, and there is no right or wrong answer. We are only interested in comparing our system's performance with a different methodology for producing segmentations.

## 2. CONDITIONS OF SUBJECT PARTICIPATION

We will not reveal your name in connection with any of our data. Data will be stored without any indication of which subject provided the data.

You may withdraw from the study at any time during your participation; if you wish us to destroy the data collected thus far from you, we will do so. If you decide to withdraw from the study, there will not be any penalties or adverse consequences. We will notify you of any significant findings that might affect your willingness to participate in this study.

## 3. RISKS AND BENEFITS

There are no potential risks to you if you participate in this experiment. If you refuse to participate, you will not be penalized. In most cases, there are no direct benefits to you if you participate in this experiment. In some cases, your course instructor may have offered you some extra course credit for your participation. If this is the case, your instructor will have clearly outlined the available amount of extra credit and the guidelines for your participation. If you do not wish to participate in the study, an alternative extra credit assignment will be provided. Under no circumstances will a course grade be penalized for your decision not to participate.

UNIVERSITY OF DELAWARE
APPROVED BY HSRB

6/12/09 - 6/11/10 EOP
DATE


_____(Please initial after reading)

If at any point during the administration of the experiment, you change your mind and wish to withdraw, you may do so and the data collected so far will be discarded. If you have received course extra credit for your participation, that extra credit will not be revoked.

4. CONTACTS

If you have any questions, or wish to withdraw from the study, please contact Dr. Sandra Carberry by calling (302) 831-1954, by sending mail to carberry@cis.udel.edu, or by writing to her at the Department of Computer and Information Sciences, 103 Smith Hall, University of Delaware, Newark, DE 19716.

If you have any questions or concerns regarding the rights of individuals who agree to participate in research, please contact the Chair of the University of Delaware Human Subjects Review Board at 302-831-2136.

5. SUBJECT'S ASSURANCES

Participation in this study is voluntary. It is your decision whether you want to participate. If you refuse to participate, you will not be penalized. If at any point during the experiment, you change your mind and wish to withdraw, you may do so. If you request to have the data collected from you destroyed, we will do so.

Experiment results and consent forms will be stored in Dr. Sandra Carberry's office to ensure confidentiality. Data will be stored for at least three years. A copy of this consent form will be provided for you.

6. CONSENT SIGNATURES

Please sign below if you agree to participate in this study:

_____ Date: _____

(Participant signature)

_____

(Please print your name)

UNIVERSITY OF DELAWARE
APPROVED BY HSRB

6/12/09 - 6/11/10 EOP

DATE



_____(Please initial after reading)

# CITI Collaborative Institutional Training Initiative

**Course In The Protection Human Subjects Curriculum Completion Report**
**Printed on 1/11/2012**

**Learner:** Peng Wu (username: pengwu.ud)
**Institution:** University of Delaware
**Contact Information** Artificial Intelligence
Data Mining
Newark, DE 19713 USA
Department: Computer and Information Science
Phone: 3024899402
Email: pwu@cis.udel.edu

**Graduate Students:**

**Stage 1. Basic Course Passed on 06/12/09** (Ref # 2888760)

| Required Modules | Date Completed | Score |
|---|---|---|
| Belmont Report and CITI Course Introduction | 06/11/09 | 2/3 (67%) |
| Students in Research | 06/12/09 | 10/10 (100%) |
| History and Ethical Principles - SBR | 06/12/09 | 2/4 (50%) |
| Defining Research with Human Subjects - SBR | 06/12/09 | 4/5 (80%) |
| The Regulations and The Social and Behavioral Sciences - SBR | 06/12/09 | 4/5 (80%) |
| Assessing Risk in Social and Behavioral Sciences - SBR | 06/12/09 | 4/5 (80%) |
| Informed Consent - SBR | 06/12/09 | 3/4 (75%) |
| Privacy and Confidentiality - SBR | 06/12/09 | 3/4 (75%) |
| Research in Public Elementary and Secondary Schools - SBR | 06/12/09 | 2/4 (50%) |
| Conflicts of Interest in Research Involving Human Subjects | 06/12/09 | 2/2 (100%) |
| University of Delaware | 06/12/09 | no quiz |
| **Elective Modules** | **Date Completed** | **Score** |
| International Research - SBR | 06/12/09 | 3/3 (100%) |

**For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and unauthorized use of the CITI course site is unethical, and may be considered scientific misconduct by your institution.**

Paul Braunschweiger Ph.D.

Professor, University of Miami
Director Office of Research Education
CITI Course Coordinator

Return