INTERACTIVE VISUALIZATIONS OF NATURAL LANGUAGE

CHRISTOPHER MERVIN COLLINS

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

Christopher Mervin Collins: *Interactive Visualizations of Natural Language*
Doctor of Philosophy

SUPERVISORS:
Gerald Penn
Sheelagh Carpendale

COMMITTEE MEMBERS:
Graeme Hirst
Ronald Baecker
Stuart Card (external)

ADVISOR NOT PARTICIPATING IN FINAL DEFENSE:
Ravin Balakrishnan

LOCATION:
Toronto

COMPLETED:
June, 2010

SUPPLEMENTARY MATERIALS:
`http://www.christophercollins.ca/thesis`

TYPOGRAPHIC STYLE:
The layout and typography of this dissertation are based on the Classic Thesis Style by André Miede.

For my grandmother Mary and in loving memory of my grandparents Paul, Minnie, and Wilfred.

They often wondered if I'd ever leave school.
With this PhD, hopefully not.

ABSTRACT

Interactive Visualizations of Natural Language

Christopher Mervin Collins
Doctor of Philosophy, 2010
Graduate Department of Computer Science
University of Toronto

While linguistic skill is a hallmark of humanity, the increasing volume of linguistic data each of us faces is causing individual and societal problems — 'information overload' is a commonly discussed condition. Tasks such as finding the most appropriate information online, understanding the contents of a personal email repository, and translating documents from another language are now commonplace. These tasks need not cause stress and feelings of overload: the human intellectual capacity is not the problem. Rather, the computational interfaces to linguistic data are problematic — there exists a *Linguistic Visualization Divide* in the current state-of-the-art. Through five design studies, this dissertation combines sophisticated natural language processing algorithms with information visualization techniques grounded in evidence of human visuospatial capabilities.

The first design study, Uncertainty Lattices, augments real-time computer-mediated communication, such as cross-language instant messaging chat and automatic speech recognition. By providing explicit indications of algorithmic confidence, the visualization enables informed decisions about the quality of computational outputs.

Two design studies explore the space of content analysis. DocuBurst is an interactive visualization of document content, which spatially organizes words using an expert-created ontology. Broadening from single documents to document collections, Parallel Tag Clouds combine keyword extraction and coordinated visualizations to provide comparative overviews across subsets of a faceted text corpus.

Finally, two studies address visualization for natural language processing research. The Bubble Sets visualization draws secondary set relations around arbitrary collections of items, such as a linguistic parse tree. From

this design study we propose a theory of *spatial rights* to consider when assigning visual encodings to data. Expanding considerations of spatial rights, we present a formalism to organize the variety of approaches to coordinated and linked visualization, and introduce VisLink, a new method to relate and explore multiple 2D visualizations in 3D space. Inter-visualization connections allow for cross-visualization queries and support high level comparison between visualizations.

From the design studies we distill challenges common to visualizing language data, including maintaining legibility, supporting detailed reading, addressing data scale challenges, and managing problems arising from semantic ambiguity.

PUBLICATIONS

Many of the ideas and figures contained herein have been previously published in the following peer-reviewed publications:

Collins, Christopher; Carpendale, Sheelagh; Penn, Gerald. Visualization of Uncertainty in Lattices to Support Decision-Making. *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, pp. 51–59, Norrköping, Sweden, May 2007. (Chapter 4)

Collins, Christopher; Carpendale, Sheelagh; and Penn, Gerald. DocuBurst: Visualizing Document Content using Language Structure. *Computer Graphics Forum (Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis '09))*, 28(3): pp. 1039–1046, Berlin, Germany, June 2009. (Chapter 5)

Collins, Christopher; Viégas, Fernanda; and Wattenberg, Martin. Parallel Tag Clouds to Explore and Analyze Faceted Text Corpora. *IEEE Symposium on Visual Analytics Science and Technology*, Atlantic City, USA, October 2009. (Chapter 6)

Collins, Christopher; Penn, Gerald; and Carpendale, Sheelagh. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. *IEEE Transactions on Visualization and Computer Graphics. (Proceedings of the IEEE Conference on Information Visualization (InfoVis '09))*, 15(6), pp. 1009–1016 , Atlantic City, USA, November–December 2009. (Chapter 7)

Collins, Christopher; Carpendale, Sheelagh. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics. (Proceedings of the IEEE Conference on Information Visualization (InfoVis '07))*, 13(6), pp. 1192–1199, Sacramento, USA, November–December 2007. (Chapter 8)

Be kind whenever possible.
It is always possible.

— Tenzin Gyatso, 14th Dalai Lama of Tibet

ACKNOWLEDGMENTS

Doctoral studies are not just another few years at school. From all-nighters at the lab to exciting travels to distant places, it has been a challenging adventure filled with ups and downs, but always surrounded by inspiring people and exciting ideas. It has been a privilege to share this intellectual and personal development with very talented researchers and teachers. They have helped me discover exciting new questions, investigate challenging problems, and explore my design interests.

The primary acknowledgement must go to my co-supervisors Sheelagh Carpendale and Gerald Penn.

Sheelagh is a talented supervisor, researcher, and mentor. I am not the only one who thinks so — during my time working with her, she was nominated and awarded the University of Calgary's "supervisor of the year" award for her outstanding skill at working with graduate students. She has an unwavering passion for research, exemplified by the large number of students she skillfully supervises, the significant grants she is awarded, and the influential papers she publishes. Over the years, she has been a valuable mentor, helping me get to know the ropes of the academic world and encouraging me to set my sights high. She has also become a dear friend. I only hope to be half the professor she is, and perhaps in the process to sleep twice as much as she does.

Gerald and I have worked together since I started my graduate studies. With his encouragement, I transitioned from core NLP research to the hybrid NLP-InfoVis work you will read about in this dissertation. Gerald is a wealth of knowledge and creative ideas, not only about our mutual computer science research interests, but about wide-ranging and fascinating topics from ancient Mayan hieroglyphics to Mexican cuisine. Gerald has been a constant support, an advocate, and has maintained a considerate ear and a positive attitude. How many people can boast that their PhD supervisor took them to Disneyland (during a 2-week research visit to California)?

Thanks are also due to my committee members, Graeme Hirst, Ravin Balakrishnan, and Ron Baecker, each of whom contributed helpful insights and maintained a focus on the rigour of the research and the need to communicate clearly. I always felt that the committee had my best interests in mind when providing advice. It was a delight and an honour to have a very well-respected member of the academic community, Stuart Card, serve as my external examiner and provide his insightful feedback.

Some of this work was conducted at IBM Research, under the supervision of Michelle Zhou, Martin Wattenberg, and Fernanda Viégas. I enjoyed the opportunity to learn about research in an industry lab setting, and to work with some of the brightest stars in the InfoVis world. I am also thankful to Kevin Knight and his research group at ISI, who invited me to work with them on the machine translation research you will read about in Chapter 7. I look forward to continuing to build on these collaborations in my new role. I am also indebted to my IBM colleagues, fellow interns, and the wider community of students in the NECTAR research network for many engaging research discussions and occasional coding advice.

This work has benefited from the day-to-day collaborations that take place in university lab settings. At the iLab I had many helpful discussions with all members of the lab, especially Petra Isenberg, Mark Hancock, Uta Hinrichs, Marian Dörk, Matthew Tobiasz, Annie Tat, Miguel Nacenta, Jim Young, and Tobias Isenberg, all of whom are good friends (and several were also good roommates). I am thankful to Saul Greenberg and Ehud Sharlin for adopting me as an honorary Calgary student. I feel this work is equally a product of both the University of Toronto and the University of Calgary, despite the title page. I also value the helpful discussions with my Toronto colleagues Vivian Tsang, Afsaneh Fazly, Afra Alishahi, Saif Mohammad, and Cosmin Munteanu.

It is important to me to personally acknowledge the people in my life who helped me get here. Thanks to my Mom, Christina, who was an unwavering support, especially during the final few months when I was writing. Her energy helped me get through this. She went above and beyond the call of duty with musical wake-up calls, never failing to help me get going, even through my most cranky mornings. Her expert proof reading also made the task of editing the dissertation a breeze. My partner Dan has been by my side through graduate school (even when he was in a different country). Dan always has the courage to tell me what I need to hear, even when I don't want to hear it. I'm thankful for his

patience while I finished this document, and I'm excited to face life's next challenges with him. I would also like to acknowledge my father, Mervin, my sister, Lesley, and my extended family and friends who have been strong supports and cheerleaders right to the finish.

This research would not have been possible without the generous support of several funding sources. Much of my dissertation research took place under the umbrella of the NSERC research network NECTAR, and I am thankful for that support. Additional support came from Smart Technologies, Alberta iCore, and the University of Toronto Awards of Excellence program. Finally, many thanks to the Banff Centre for providing me a peaceful studio in which to write.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ACRONYMS

CL	computational linguistics
NLP	natural language processing
MT	machine translation
InfoVis	information visualization
ASR	automatic speech recognition
IR	information retrieval
DOI	degree-of-interest
WSD	word-sense disambiguation
KWIC	keyword-in-context
API	application programming interface
HCI	human-computer interaction
DH	digital humanities
RSF	radial space-filling

COMMON TERMINOLOGY

CASUAL INFOVIS The use of computer-mediated tools to depict personally meaningful information in visual ways that support everyday users in both everyday work and non-work situations (Pousman *et al.*, 2007).

COMPUTATIONAL LINGUISTICS The scientific study of language from a computational perspective, interested in providing computational models of various kinds of linguistic phenomena.

GLYPH A graphic symbol (shape) whose appearance conveys information (also called a grapheme).

IDEOGRAPHIC Writing systems using ideograms — graphic symbols representing ideas or concepts.

INFORMATION GRAPHIC An illustration of data in a representation useful for analysis.

INFORMATION VISUALIZATION Computer-supported, interactive, visual representations of abstract data to amplify cognition.

LINGUISTIC VISUALIZATION The presentation of linguistic data through visual representations designed to amplify cognition or communicate linguistic information.

LINGUISTIC VISUALIZATION DIVIDE The gulf separating sophisticated natural language processing algorithms and data structures from state-of-the-art interactive visualization design.

LOGOGRAPHIC Writing systems using logograms — graphic symbols representing words or morphemes.

LOGOSYLLABIC Writing systems in which the glyphs represent morphemes, which, when extended phonetically, represent single syllables.

MULTI-WORD COLLOCATE Sequential words treated as a single semantic unit.

NATURAL LANGUAGE PROCESSING Linguistic algorithms for applied purposes, such as summarization, translation, and speech recognition.

ONTOLOGY A rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations.

PHONETIC Writing systems which directly represent the sounds of spoken language.

PICTOGRAPHIC Writing systems using pictographs — graphic symbols that more or less portray the represented object.

SEMANTIC ZOOM An interaction operation in which the semantic level-of-detail of visible data changes, as opposed to geometric zoom, which is a simple geometric scale operation on the current view.

SPATIAL RIGHTS The assignment of position as a visual encoding for a data dimension or relation.

VERNACULAR VISUALIZATION The use of "unsophisticated" visualizations by non-experts to "get things done" (Viégas and Wattenberg, 2008).

WALK-UP-AND-USE Interfaces and interaction techniques which provide affordances that negate the need for instructions.

Part I

ON VISUALIZING LANGUAGE

1

INTRODUCTION

One picture is worth ten thousand words.

— Royal Baking Powder Advertisement, 1927[1]

Linguistic skill is a hallmark of humanity, setting us apart from our fellow hominids; we can express ourselves with nuance and power through many forms, such as fiction, poetry, music, and political orations. From infancy, we progress through stages of development, refining our communication skills, learning to speak, listen, read, and write. We also learn to make and interpret physical gestures, using actions to augment our words. Human linguistic capability is truly astonishing: the communication abilities of a young child surpass the abilities of any other known form of life (Pinker, 1997). By adulthood, we thrive in an environment of near constant communication in work and personal life (Janusik and Wolvin, 2009). The media of communication are in constant flux. Recent years have brought increased free and open access to ever more information through the Internet. While this is an exciting development, managing, exploring, and analysing the flow of linguistic and related data is becoming an individual and a societal problem. This familiar problem is often called *information overload*, a term originating in the psychological studies of Miller (1960, 1962, 1964) and later popularized in the writings of Toffler (1971). The problem of information overload should not, however, be attributed to human intellectual deficiencies — every individual is different, and there is much variety in the volume and form of content one can manage comfortably. Rather, the information is overwhelming because of the naïve *manner* in which it is delivered. A scholar could spend a lifetime reading in a specific subject area and retire without scratching the surface: on top of the volume of scholarly works to carefully read, she would spend much of her time keeping abreast of email, blogs, social network updates, work-related memos, reviewing requests, news media, and student reports. Today's pace of creation and dissemination far sur-

information overload

1 This well-known quote is often incorrectly attributed as a "Chinese Proverb". Fred Barnard of Royal Baking Powder thought this false provenance would bring a feeling of authenticity to the phrase (Mieder, 2004, p. 82).

passes our ability to consume information using traditional means such as sequential reading (Klingberg, 2008, p. 5).

Computers can process bulk data quickly, building models encapsulating large volumes of information. Could we solve the overload problem by offloading decision-making involving linguistic data to computers? Assuming people would want such a solution (a doubtful prospect), it is unlikely to be possible in the foreseeable future. A child's language and reasoning skills easily surpass current computational capabilities. Without convincing artificial intelligence, computers lack the subtlety of pattern recognition, the ability to see exceptions to the rule, and the ease of understanding content in the context of world knowledge that comes so naturally to us. If Searle (1980) is correct, we may never reach the point of autonomous computing capable of performing these tasks as well as a human. In other words, for the foreseeable future, people will remain an integral part of the linguistic information analysis process.

There is, however, a possible *computer-mediated* solution. Beyond linguistic skills, humans also possess strong *visuospatial reasoning* abilities which can be leveraged in conjunction with language skills (Tversky, 2004). Thus one potential way to design human-computer optimizations for language data is to use computers to process bulk data and present it in forms that are interpretable using our visuospatial abilities. Information visualization (InfoVis) is a field of research dedicated to designing and studying interactive visual representations of abstract data. The topic of this dissertation is bringing the advantages of natural language processing (NLP) and InfoVis interfaces to bear on the linguistic information overload problem. The following chapters present five design studies which are all examples of closely coupled NLP algorithms with InfoVis techniques well motivated by our understanding of human visual information processing capabilities and interaction preferences.

1.1 UNDERSTANDING THE SPACE OF LINGUISTIC VISUALIZATION

Natural language data provide some unique challenges for visualization: text consists of abstract concepts that are represented in many different ways, the data are often nominal (unordered), ambiguity in meaning is built into the data, and much of the semantic interpretation depends on context and common cultural understanding (Hearst, 2002). Consider the difficulty in automatically visualizing the nuanced meanings of words,

social and psychological concepts, and causal relationships. Language has a very high dimensionality, and many subsets of dimensions can be combined together, compounding the problem. Furthermore, algorithmic manipulations of language may introduce uncertainty and error when transforming raw data into analytical abstractions useful for visualization.

space of linguistic visualization

This dissertation explores several ways to frame the *space of linguistic visualization*, arguing for close coupling of linguistically sophisticated data processing with well-designed InfoVis. In order to inform our work, we examine visualizations of language created by a variety of scholarly, artistic, and professional communities. However, primarily, this dissertation explores a range of application areas to which NLP has been previously applied and in which visualization may be beneficial, from document content summarization to machine translation. Paralleling the progression of problem areas, the design studies target a range of audiences: Internet chat participants, library users, legal scholars, linguistic researchers, and professional data analysts.

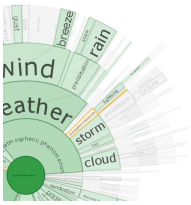
The examples simultaneously span a range of NLP approaches, including word counting methods, statistical algorithms, and drawing on knowledge resources authored by linguistic experts. They also cover a range of data types as their basic unit of visualization: part-of-speech tagged words, multi-word collocates, sentences and sentence fragments, and structural overviews of lexical ontologies and document corpora. The structures we propose for organizing the space of linguistic visualization are further explored in Chapter 3 but have been mentioned here to frame the selection of design studies introduced in the following subsection.

1.1.1 *Five Design Studies*

These design studies were selected to provide a broad coverage across the above mentioned dimensions. Each of the design studies explores a different area of linguistic application, targets a different audience, and draws on NLP algorithms and data structures of a variety of types. In addition to exploring the space of people, algorithms, data structures, and application areas, these studies also expand on the design decisions and considerations made in the creation of visual encodings, spatialization methods, and interaction techniques. From the lessons of these design exercises, we distill some considerations specific to working in the interdisciplinary realm of linguistic visualization.

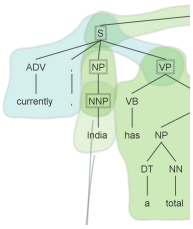
The five prototypes we present are Uncertainty Lattices, DocuBurst, Parallel Tag Clouds, Bubble Sets, and VisLink.

UNCERTAINTY LATTICES The Uncertainty Lattices is linked to a phrase-based statistical machine translation (MT) system, and is targeted at members of the general public who use language technologies such as MT and automatic speech recognition (ASR). It is an example of opening up the black box of NLP to reveal new and potentially helpful information using visualization methods designed to bring the reader into the decision loop.

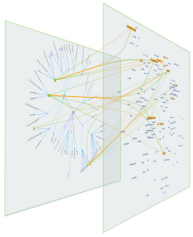


DOCUBURST DocuBurst is a document content visualization which uses the words and multi-word collocates within the lexicographer-created ontology *WordNet* (Fellbaum, 1998) to create visualizations of document content in which relative spatial position carries semantic significance. This visualization is targeted at people seeking overviews and comparisons of long documents, for example patrons searching a library collection.

PARALLEL TAG CLOUDS Parallel Tag Clouds operate on the word and collocate level, and were developed for providing legal scholars access to overviews which differentiate the language of one court from another, and one time period to another. The outcomes are applicable to investigations of the internal variety within any large, faceted text corpus.



BUBBLE SETS Bubble Sets is the result of a collaboration with MT researchers working with syntax-based MT. It is designed partially based on statistical models of a trained MT system, but has capacity to include hand-created data and annotations by linguistic experts. The visualization technique developed for this work is readily generalizable to a number of linguistic and non-linguistic information domains.



VISLINK Finally, VisLink introduces a new way to link multiple heterogeneous 2D visualizations in a 3D space. Data units can range from lexical items to entire document corpora, and any type of linguistic algorithm can be connected. VisLink can reveal previously unseen relations between any number of linguistic (and non-linguistic) visualizations. Its complexity and power suggests that the most appropriate audience would be data domain experts and researchers.

1.2 LANGUAGE REPRESENTATIONS THROUGHOUT HISTORY

Human communication is inherently visual — visual representations often accompany text and speech. We use visual cues such as gestures to clarify ambiguous speech. We rely every day on symbols and signs to stand in for written language as we find our way around our cities. We recognize the inadequacy of plain alphabetic text for emotionally rich dialogue, and embellish our instant messaging conversations with various emoticons (Hancock *et al.*, 2007). It is not only today's society that communicates visually. To place our discussions of recent efforts to develop interactive visualizations of language (several examples of which compose this dissertation) into context, it is informative to reflect on a brief history of language representations throughout history.

Discoveries at Lascaux, France reveal the visual representational practices of very early humans. Sequential paintings on cave walls, created around 16,000 BCE, are thought to be for ritual-teaching purposes (see Figure 1.1). Such pictographic writing systems visually recall the concepts represented. At this stage in history, there was no distinction between *reading a text* and *describing a picture* — saying the same thing meant conveying the same meaning, not using the same words (Olson, 1996a). Around 3,200 BCE Sumerian logographic writing began to emerge in Mesopotamia with about 2,000 signs. By 2,600 BCE, the writing system had begun to evolve into Old Assyrian cuneiform, a logosyllabic script containing a mix of several hundred phonograms and determinatives, many logographic in origin (Horn, 1999).

Two centuries later, Egyptians were using a fully developed mix of phonetic and ideographic writing and artwork in large information murals in their temples (see Figure 1.2) (Manske, 2000). Other early visual languages, including other forms of Rebus writing showed a progression toward representing spoken language directly, leading to the creation of alphabets, which abstract the sounds of a spoken language into smaller units, and are visually decoupled from semantics. The driving forces of this progression are the subject of some disagreement: Havelock (1982, p. 11) argues for a four-stage linear evolution, from pure logographic depiction to phonetic alphabets at the pinnacle; Olson (1996a) disagrees with this evolutionary progression of writing as an ever-more-precise representation of spoken language, arguing that writing systems devel-

FIGURE 1.1: Ancient paintings in the caves of Lascaux. Visual forms may have served for utilitarian communication of rituals (Šantić, 2006). *Reprinted with permission.*

oped to precisely record *meaning*, and the correspondence to speech is a by-product.

Eventually alphabetic writing and visual representation became separate domains, with visual representations used alongside alphabetic writing; for example, illustrations often accompany text. There are exceptions in the chronological progression toward alphabetic systems: pre-contact North American Ojibwe used picture-writing systems on birch bark scrolls (*Wiigwaasabak*) dating back to about 1570 CE (Kidd, 1965, 1981). Chinese calligraphy is still a visual ideographic language, with symbols representing entire concepts. Phonetic and logographic systems are powerful visualizations, and both require time and effort to learn. However, once they are mastered, they are incredibly flexible and expressive.

Mass production of writing was restricted by manual labor throughout most of human history — first carving stone tablets, later painstakingly copying manuscripts with a quill. With the advent of ever greater means to disseminate written language, such as the printing press and proliferation of paper, visual representations of language have become ever more prevalent, and indeed have changed the course of cultural evolution (Olson, 1996b). The growth of the availability of printed text prompted the philosopher Denis Diderot, in his monumental *Encyclopédie, 1755*, to write:

FIGURE 1.2: Cursive writing on the Papyrus of Ani. Egyptians were the first to produce illustrated manuscripts with words and pictures to communicate information. *Image in the public domain.*

> As long as the centuries continue to unfold, the number of books will grow continually, and one can predict that a time will come when it will be almost as difficult to learn anything from books as from the direct study of the whole universe. It will be almost as convenient to search for some bit of truth concealed in nature as it will be to find it hidden away in an immense multitude of bound volumes (as translated in (Diderot *et al.*, 1964, p. 299)).

This remark is as true today as it was 250 years ago. Computer technology and the Internet have facilitated exponential growth in our capacity to create, access, and archive text. Despite the continued dominance of speaking and listening for daily communication, we are increasingly abandoning voice-based communication in favour of wikis, email, blogs, and SMS messages for everyday personal and professional communication (Janusik and Wolvin, 2009, p. 117). Face-to-face communication now accounts for less than half of college students' time spent communicating (Janusik and Wolvin, 2009, p. 115).

Textual data is at the forefront of information management problems today. Millions of pages of text data, in many languages, are produced daily: emails, news reports, blog posts, product reviews, discussion forums, academic articles, and business reports. As an example, in 2006, Google released an *N*-gram corpus built by scanning over one trillion words of Web text (Franz and Brants, 2006), and this was not even the full extent of publicly available text at that time. Since 2006, the number of active websites has more than doubled: the Web now hosts over 235 million hostnames, and more than 70 million active sites (Netcraft, 2009). In addition to this, the technological innovations opening up the Web as a programming platform for freely sharing data (*Web 2.0*) have made social data, news media, governmental data, books, historical records, call transcripts, product reviews, and other sorts of information easily accessible with open APIs (*e.g.*, Netcraft, 2009; New York Times, 2009; Open Library, 2009; Sunlight Foundation, 2009; Twitter, 2009; Wattenberg *et al.*, 2008).

We are once again changing the way we read, moving toward electronic libraries and e-books. As we are faced with large amounts of textual information, plain text representations and poorly designed interfaces may be exacerbating our experience of information overload. It is now an appropriate time to systematically revisit visual depictions as a form of enhancing linguistic communication — the topic of this dissertation.

1.3 THE LINGUISTIC VISUALIZATION DIVIDE

The ability of computers to quickly process electronically encoded linguistic data has lead to a wealth of important NLP tools to handle the increasing data flow. Automatic summarization, MT, ASR, question answering systems, and information retrieval (IR) tools have become part of everyday life. There has also been an explosive growth in the variety of ways language is visually represented. New visualization techniques appear on the Internet almost daily (Moere, 2009, provides a updated listing). These tend to focus on word counting, and generally represent language using visual manipulations of existing writing systems (scaled text, colour variations, meaningful spatial arrangements of words). There is a notable lack of collaboration between those with expert knowledge in NLP and those designing new interactive visual representations. Through closer integration of NLP and InfoVis we can create interactive visual

representations of language that may provide useful new perspectives on the mountains of electronic text inundating us every day.

When this dissertation research was started in 2004, the field of text visualization could be summarized in a handful of influential contributions (*e.g.*, Havre *et al.*, 2002; Hearst, 1995; Hetzler *et al.*, 1998a; Paley, 2002; Wise *et al.*, 1995). At that time, the research focus was on visualization to support IR. In fact, visualizations targeted at providing overviews for IR have largely been difficult to use and development in this area has slowed (Hearst, 2009, ch. 10). With new attention on visual analytics and sense-making (Thomas and Cook, 2005), attention turned to providing overviews of large corpora for specific purposes: *e.g.*, evaluating emotional content (Gregory *et al.*, 2006) and named entity occurrences (Stasko *et al.*, 2007; Weaver *et al.*, 2007). Since that time, the InfoVis community has begun to realize the potential of linguistically-driven visualization: there are now annual sessions on text analytics at several major conferences, including *IEEE InfoVis*, *IEEE VAST*, and *EuroVis*.

Visualization has been used as a tool to analyze language from many perspectives: linguistic (Kempken *et al.*, 2007), humanistic (Plaisant *et al.*, 2006), forensic (Stasko *et al.*, 2007), historic (Weaver *et al.*, 2007), and sociologic (Harris and Kamvar, 2006; Viégas and Smith, 2004; Viégas *et al.*, 2004). Thomas and Cook (2005) cite large-scale text visualization as one of the *grand challenges* of the newly defined field of Visual Analytics, while Chen (2005) regards the scalability of text visualization as one of the top ten open research problems. Indeed, the nascent and growing field of digital humanities (DH) (Schreibman *et al.*, 2004) is built upon applied NLP, and occasionally turns to visualization as a means to enhance scholars' ability to understand large bodies of text (*e.g.*, Don *et al.*, 2007; Mueller, 2008; Sinclair, 2006; Sinclair and Rockwell, 2009). Approaching linguistic analysis with a combination of text processing, information visualization, and human decision making, is an exciting and active field.

text visualization: a grand challenge

Most of the recent works visualize the *end outputs* of a text analysis system. For example, a tag cloud of previously counted words, or a treemap of detected concepts. Some of these restrictions may be due to limitations of the Web as a computing and visualization platform. In other cases, the linguistic analysis is intentionally simplistic, to appear transparent to viewers (Feinberg, 2008). Often, the fact that the data is text-based and coming from an NLP process seems to be of no consequence

for the visualization design, mirroring Kirschenbaum's critique of the traditional software development cycle:

> Yet the truth is that, from a developer's perspective, the interface is often not only conceptually distinct, but also computationally distinct. Like the history of hand-press printing, which teaches us that paper-making, typesetting, etching or engraving, and bookbinding came to encompass very different domains of labor and technical expertise (rarely housed under the same roof), resource development in the digital world is typically highly segmented and compartmentalized (Kirschenbaum, 2004).

We call this the *linguistic visualization divide*: "The gulf separating sophisticated natural language processing algorithms and data structures from state-of-the-art interactive visualization design.". It is evident in many published linguistic visualizations. Linguistic researchers spend entire careers developing models of how languages are structured (syntax, semantics), theorizing about how the human mind learns and recalls linguistic knowledge, and creating computational models to perform linguistic manipulations on data (*e.g.*, translation, natural language understanding). Yet, interactive visualizations of language developed by visualization researchers to support particular analysis tasks are often lacking linguistic sophistication beyond word counting (*e.g.*, Havre *et al.*, 2002; Rohrer *et al.*, 1999; Viégas *et al.*, 2009).

linguistic visualization divide

Conversely, visual representations of language developed by computational linguists are often simple information graphics used to present research results (we explore examples in Chapter 3). By our own informal review of the *Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2006*, 42% of the papers contained at least one novel (custom-designed) information graphic, excluding line and bar charts, parse trees, and process diagrams. That was a suprisingly high result, and led us to begin to realize that visualization was happening *in the wild*, in research communities not versed in the theory and practice of interactive visualization. Possibly it happens out of a necessity to cope with large volumes of data.

visualization in the wild

Rarely are interactive visualizations used to provide direct access to both the ongoing processes and the outcomes of linguistic analysis. Examples bridging the divide, providing linguistic sophistication to leverage computational power, and interactive interfaces informed by studies of

human perception, are rare. The examples presented in this dissertation begin to bridge this divide in that the designs are informed by both linguistic and visualization expertise, and the visualization algorithms are coupled directly with real, on-line natural language processes, such as statistical translation, keyword detection, and parsing.

Of course, to truly bridge this divide, the linguistic algorithms and visualization would be developed simultaneously, to mutual benefit, whereas this work leverages existing research in NLP to create custom-designed visualizations. The design studies in the following chapters serve as motivation for future simultaneous development of linguistic analysis and visualization algorithms. These examples explore the potential advantages of interactive visualization through re-appropriation of existing algorithms and data structures for new uses.

1.4 THESIS PROBLEM AND APPROACH

This dissertation will investigate whether linguistic data and algorithms, as designed for computational linguistics (CL) and NLP purposes, can be re-purposed to drive interactive visualizations of language. The goal of creating such linguistically-integrated visualization is to reveal the complexities of linguistic processes and data that have been previously hidden within the computational processes. Example tasks include finding the most appropriate text to read, quickly understanding the contents of a large corpus, and being aware of potential errors in computer-manipulated texts.

The forms of visual depiction proposed here are not *logographic*; they are not directly mapped to physical reality or abstract ideas (although some attempts at this sort of linguistic visualization have been made (*e.g.*, Coyne and Sproat, 2001)). The method we employ is to take linguistic algorithms and data of various forms and create structured, meaningful, and interactive visual forms to enhance interpretation of the data, or to reveal information previously not visible with standard techniques. It is important to note that the goal is to *augment*, not replace, the act of reading.

Our approach involves first exploring the space of ongoing challenges in NLP and information management, and, guided by a literature review, mapping several dimensions on which coupled NLP–InfoVis can be placed. We target a variety of common application areas in which NLP is

currently in use, examine the algorithms and data structures, and identify deficiencies in current representations which may be enriched by application of InfoVis. Our InfoVis research is grounded in theories and evidence about human cognitive processing and perceptual capabilities (Clark and Chalmers, 1998; Stone, 2003; Tversky, 2004; Ware, 2004; Wolfe, 2003), design practices developed and refined by a community of researchers (Amar and Stasko, 2004; Bertin, 1983; Carpendale, 2003; Cleveland and McGill, 1984; Tufte, 2001; Zuk and Carpendale, 2006), and carefully-selected or custom-designed interaction techniques to allow the manipulation and exploration of the visualization (Card *et al.*, 1999; Shneiderman, 1996; Yi *et al.*, 2007).

Where appropriate, we formalize a general problem area as it relates to InfoVis, and describe how our new techniques expand the range of possible representations. Each of the design studies demonstrates a way to see and interact with linguistic information that was not previously possible.

1.5 CONTRIBUTIONS

The contributions of this dissertation fall into three areas: new visualizations for language data, generalizable technical innovations, and high-level visualization concepts.

Our prototypes each contribute new techniques to visualize language. The Uncertainty Lattices visualization exposes the *black box* of statistical MT and ASR, to help people to make decisions about the quality of the outputs (Chapter 4). DocuBurst presents the first document content visualization based on an expert-created structure of language (Chapter 5). Parallel Tag Clouds (Chapter 6) present a method for displaying and exploring differences within very large faceted text corpora. Bubble Sets address a specific and repetitive analysis task important to a group of MT researchers (Chapter 7). Finally, VisLink provides a general platform within which multiple visualizations of language (or other data types) can be connected, cross-queried, and compared (Chapter 8).

Generalizable visualization and interaction techniques arising from this research include:

- two methods for encoding uncertainty in the background of graph nodes (§4.5)

- a method for expanding radial space-filling (RSF) trees using a mouse wheel (§5.5.3)
- a method for representing connections in multiple tag clouds (§6.3)
- a method for pre-computing, storing, and dynamically composing lists of significant words at interactive speeds (§6.5)
- a method combining obstacle avoidance and implicit surfaces to draw concave set boundaries at interactive speeds (§7.3)
- a method for interactively assigning set membership without need for proximity (§7.3.5)
- a method to re-use the spatial visual dimension by displaying multiple 2D visualizations in a 3D space (§8.2)
- interaction widgets for using a mouse to manipulate 2D planes in 3D space (§8.2.2)

Theoretical contributions arising from this work include a definition of and exploration of the concept of *spatial rights* (§9.2.3), a formalism to describe the space of multiple relationship visualizations (§8.1), and the concept of cross-visualization queries (§8.2.4).

1.6 STRUCTURE OF THE DISSERTATION

This dissertation is framed around a series of visualizations of language built upon a variety of linguistic technologies and addressing a variety of target audiences. Before presenting the design studies, two background chapters review the design influences behind this research then investigate and organize the space of related work in linguistic visualization. Following this background, our five design studies are divided into three parts by problem area and target audience. Particular attention is paid to exploring the design process, describing alternatives considered and decisions made to best support particular tasks on the data.

1.6.1 *On Visualizing Language*

Following this introduction and motivation chapter, Chapter 2 introduces the power of InfoVis as a tool for analysis, and describes the effectiveness of visualization as a cognitive aid (§2.2). Moving forward, we review the design theories which, while applied with flexibility, have been influential in the projects presented in this dissertation (§2.3).

Chapter 3 begins with an examination of the variety of communities creating linguistic visualizations. Focusing more specifically on contributions from NLP and InfoVis, the remainder of the chapter reviews examples of related work structured around an exploration of the ways in which the *space of linguistic visualization* can be organized.

The following three parts of the dissertation present design studies which exemplify the types of NLP tasks and target audiences which we have investigated with visualization. As the design studies are discussed, additional, area-specific, related work will be reviewed. Where appropriate, we will give examples of how visualization algorithms created for linguistic purposes can generalize to address other data domains.

1.6.2 *Visualization for Real-time Communication*

Part II presents a visualization technique to reveal multiple hypotheses and meaningful confidence scores for MT and ASR (Chapter 4). The approach provides a *human-in-the-loop* decision making process by visually exposing the data structures and confidence scores used by an MT system.

1.6.3 *Visualization for Content Analysis*

Part III reviews two examples of interactive visualization designed to provide overviews and drill-down interaction for examining the contents of large amounts of text. Chapter 5 outlines the DocuBurst project, an example of bringing together visualization research (RSF trees and saturation colour scales) and language structures designed by experts (*WordNet* hierarchy) for mutual benefit, allowing for overview and comparison of the contents of books.

Chapter 6 presents Parallel Tag Clouds, a text analysis technique for large, faceted, document corpora based on text extraction techniques from summarization and IR, and visualization techniques from parallel coordinates plots (Inselberg, 1985).

1.6.4 *Visualization for Linguistic Research*

Part IV moves beyond visualizations prioritizing simplicity of form and ease-of-interaction to introduce visualizations which, while being more

complex, provide analytical capabilities appropriate for natural language engineers and linguists.

Chapter 7 first describes our collaboration with NLP researchers at work (§7.1). Next, Section 7.3 describes the prototype visualization and finally Section 7.5 describes how Bubble Sets are generalizable to several problem domains.

Chapter 8 will describe the VisLink concept, a platform for interactively linking multiple 2D visualizations in 3D space through the reuse of the spatial *visual variable*. After formalizing the space of multi-relation visualizations (§8.1), this chapter describes the design of and interaction with visualization planes (§8.2). VisLink serves as a bridge, a system for linking arbitrary pairs of visualizations, for example a force-directed graph and a radial graph. Interaction provides the ability to form rich, cross-visualization queries. Section 8.4 shows how the technique can be generalized to non-linguistic data.

1.6.5 *Closing*

Chapter 9 recounts common challenges arising from our experiences designing visualizations of language (§9.1), summarizes the contributions of this dissertation (§9.2) and outlines promising directions for future investigation of close coupling of interactive visualization and NLP algorithms which bridge the linguistic visualization divide (§9.3).

2

THE POWER OF INFORMATION VISUALIZATION

Computers are useless. They can only give you answers.

— Pablo Picasso, 1968

Information visualization is the research field investigating "Computer-supported, interactive, visual representations of abstract data to amplify cognition." (Card *et al.*, 1999, p. 7). In amplifying cognition, information visualization (InfoVis) can support exploration, reveal hidden patterns, raise new questions, improve analytical efficiency, and reduce cognitive strain. Hegarty (2004) has explored how external visualization ("diagrams in the world") and internal visualization ("diagrams in the mind") relate; the process of interpreting InfoVis appears to use both types. InfoVis and 'visualization' in this section (and the dissertation in general) refer to *external visualization*. Ware (2004) defines five advantages of effective information visualizations:

COMPREHENSION Visualization provides an ability to comprehend huge amounts of data.

PERCEPTION Visualization reveals properties of the data that were not anticipated.

QUALITY CONTROL Visualization makes problems in the data (or in the data collection methods) immediately apparent.

FOCUS + CONTEXT Visualization facilitates understanding of small-scale features in the context of the large-scale picture of the data.

INTERPRETATION Visualization supports hypothesis formation, leading to further investigation.

The field is founded on evidence that the way data is represented can affect our ability to use it for a specific task. For example, cognitive science studies have revealed that using our high-bandwidth visual system to learn and remember diagrams can be as much as twice as efficient as learning the same information with plain text (Landauer, 1986). The use of external aids to help solve cognitive problems is often called *external cognition*. After describing some of the challenges motivating the search for cognitive aids (§2.1), Section 2.2 will review some theories of external

cognition relevant to InfoVis. Driven by the potential for visualization to increase cognitive capacity, Section 2.3 reviews the significant theories and design frameworks influencing the design of the prototypes presented in later chapters.

2.1 CHALLENGES OF AN INFORMATION SOCIETY

Being overwhelmed by information is not a trivial issue given the importance of information (much of it linguistic in nature) in an *information society*. Despite contemporary use of the term, an information society means more than "we all use the Internet now". Indeed, the term is credited to Machlup (1963), writing in a world without the Internet. While there is some controversy on the precise definition, there is general consensus that it includes the notions of a society where information carries primary economic importance, and a large proportion of the population are employed as *knowledge workers* (Webster, 2006). Wark (1997, p. 22) describes the habitual means of transfer of information in an information society (talking on the telephone, sending mail) as "second nature". He introduces the concept of "third nature", a technologically-augmented extension of second nature which is able to "speed up, proliferate, divide, mutate, and beam in on us from elsewhere" (Wark, 1997, p. 25), and in the process, stress us with information overload. While in the intervening years since Wark's argument we have become more accustomed to technologically-augmented communication, the ever increasing volume has prevented it from truly becoming as comfortable as second nature.

The information volume problem goes beyond a need to filter and retrieve the right information: the lists of retrieved results are often too long to read in entirety. For example, in August 2009, an Internet search for a relatively obscure (and recent) term, 'DocuBurst' (the name of the project described in Chapter 5) yielded 2,690 results. Of course, this term has been used for other purposes in the past, but the majority of the sites are about our project: published papers, blogger praise and criticism, press coverage, and academic citations. In a world where everyone with access to computers and Internet technology (*i. e.*, those of us on the access side of the *digital divide* (Warscchauer, 2004)) can self-publish, too much is written on just about every topic to simply read it all. We need help with integration and abstraction. In Figure 2.1 we illustrate this for linguistic data with an example of visualization usage moving from no external

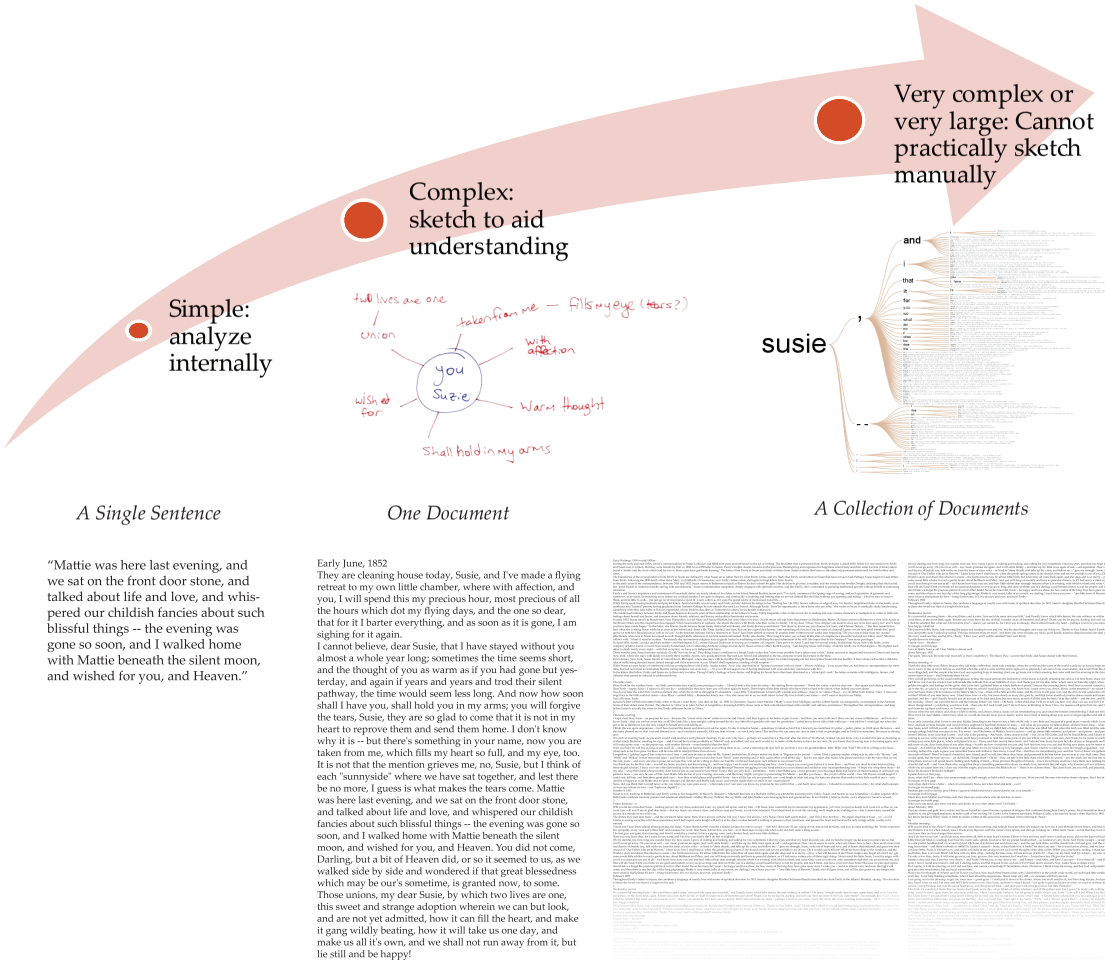


FIGURE 2.1: Increased complexity motivates the use of external cognitive aids (Norman, 1993). Here we see an increasingly large amount of data from the collection of Emily Dickinson's letters to her sister-in-law Susan Gilbert. There is intense interest in the nature of their relationship, and InfoVis has been applied in the past to this data (Plaisant *et al.*, 2006). From left to right: Analyzing a single sentence could be simply done by reading and pondering it. Understanding relations expressed in a single letter could be supported by a sketch, but to understand the use of "Susie" in the entire collection, we may turn to an interactive visualization such as a Word Tree (Wattenberg and Viégas, 2008).

aid, to a manually created sketch, to an interactive visualization as the complexity and volume of linguistic data grows.

2.2 COGNITIVE AIDS

This section briefly reviews the cognitive science theories of enhanced cognition: cognitive advantages through synergies between human abilities and the external supports we use. This background is intended as a motivation for why this dissertation turns to visualization as a possible means for reducing overload and enhancing insight about linguistic data.

Humans are tool makers and tool users by nature. From stone-age arrowheads to the latest mobile communications device, we have a long and intricate relationship with external objects to improve our lives. We are adept at 'seeing' an object differently depending on our needs: a stick may be fuel for a fire, a cane for walking, a weapon, a building material, or simply a broken tree branch lying on the forest floor. Our relationship to, and understanding of, external objects changes with the context. Beyond physical aids, we create and use tools and techniques as "external aids that enhance cognitive abilities" (Norman, 1993, p. 43). We are all familiar with using aids to jog our memories (to do lists, mnemonics), to learn the meanings of unknown words (dictionaries), and to help us perform mathematical calculations (pen and paper). These are not just helpful tools, but are actually ways to extend our cognitive abilities beyond that of an un-aided mind (Clark and Chalmers, 1998).

Evidence of increased cognitive capacity and speed through tool use range from the advantages provided by nautical slide-rules (Hutchins, 1995), to the speed gain of using pencil and paper for long multiplication (Card and Mackinlay, 1997). While there is not complete agreement on the name of this phenomenon — distributed cognition (Hutchins, 1995; Zhang and Norman, 1995), external cognition (Card and Mackinlay, 1997; Scaife and Rogers, 1996), or extended mind (Clark and Chalmers, 1998) — in all cases, the tools we use to enhance our abilities are called *cognitive aids*.

2.2.1 *Representational Change*

Changes in the way information is represented can bring about surprisingly pronounced changes in the types of tasks one can do with it. Here we define *representation* as "a formal system or mapping by which the information can be specified" (Marr, 1982). Simon and Hayes (1976) introduced the power of representational change in their experiments with *problem isomorphs* — different expressions (representations) of the same problem. When subjects were presented isomorphs of the Tower of Hanoi puzzle, expressed in a variety of ways, significant effects on speed and accuracy of solutions were found. Minor changes in the framing of the problem as a 'change' problem or a 'transfer' problem seemed to influence how the participant internally represented the problem (*i. e.*, their *mental model*).

representations support tasks

A second classic example of the power of representation change is that a number, such as 77, can be easily multiplied and counted in base-10 Hindu-Arabic numerals.[1] We have a systematic way to use pen and paper to carry out long multiplication using that representation. If we represent the number in Roman numerals LXXVII, multiplication becomes a challenge, but if we want to add, we simply append to the end and follow a set of aggregation rules (Marr, 1982). Zhang and Norman (1995) expand on the power of representational change through an investigation of numeration systems, and propose a model of distributed numerical cognition to explain the task-specific advantages of certain representations.

Representational change also occurs in language. Chapter 1 introduced a brief history of written representations of language. Particular representations may be better suited for different tasks. We can imagine ideographic cave paintings to be useful for depicting important survival skills such as avoiding dangerous animals, and for recording ritual practices or historic events. Phonetic writing systems may struggle to capture the nuance of a scene, but are quite capable of expressing abstract or even metaphorical ideas such as "visualization is opening a window onto a world of data". Furthermore, we could represent the same words as speech — a waveform played back will allow a listener to easily interpret what was said. A spectrogram of the speech may be useful for a speech therapist. In the realm of publishing, representation changes in the form

1 We choose to use the name Hindu-Arabic to acknowledge the influence of both civilizations on the development of the system most commonly known as Arabic numerals in Western countries.

of typography and text illumination also have a long history of "extending the meaning of a written text" (Drucker, 1984).

2.2.2 *Information Graphics*

Visual representations are one powerful form of cognitive aid. Representational changes to support task-specific analysis form the foundation of information graphics. An information graphic is "An illustration of data in a representation useful for analysis." (Tufte, 2001). Information graphics of data in areas such as economics and statistics date back to at least the 16th century. Information graphics to interpret text are not as common as the statistical charts, time plots, and maps described by Tufte (2001). The interpretative Biblical charts of Larkin (1918) are an early example of combining illustration, text labels, and references to sections of source text, in a spatialized narrative structure to present the author's interpretation of a book (see Figure 2.2).

Many different representational methods are used to visually encode information in information graphics such as those of Larkin (1918). Tversky (2004) presents a survey of the evidence of the extents of human *visuospatial capabilities*: spatial position, rotation, distortion, scale, and explicit connections between elements are some of the visual manipulations that can be incorporated into our reasoning about a situation. Larkin and Simon (1987) show that our ability to read distance, direction, and other spatial properties from information graphics help us to understand diagrammatic representations faster and solve problems with greater accuracy than when presented with sentential representations of the same information.

2.2.3 *Information Visualization*

The differences between interactive information visualizations and static information graphics include the ability for display changes and animation, thus allowing perceptual monitoring, and the manipulability of the medium, which allows for exploration of the information space. Information visualization, through interactive exploration, engages people with data more than a static graphic does. This engagement can support *knowledge crystallization tasks* — active analysis processes in which one gathers information, organizes it to make sense of it, and packages it

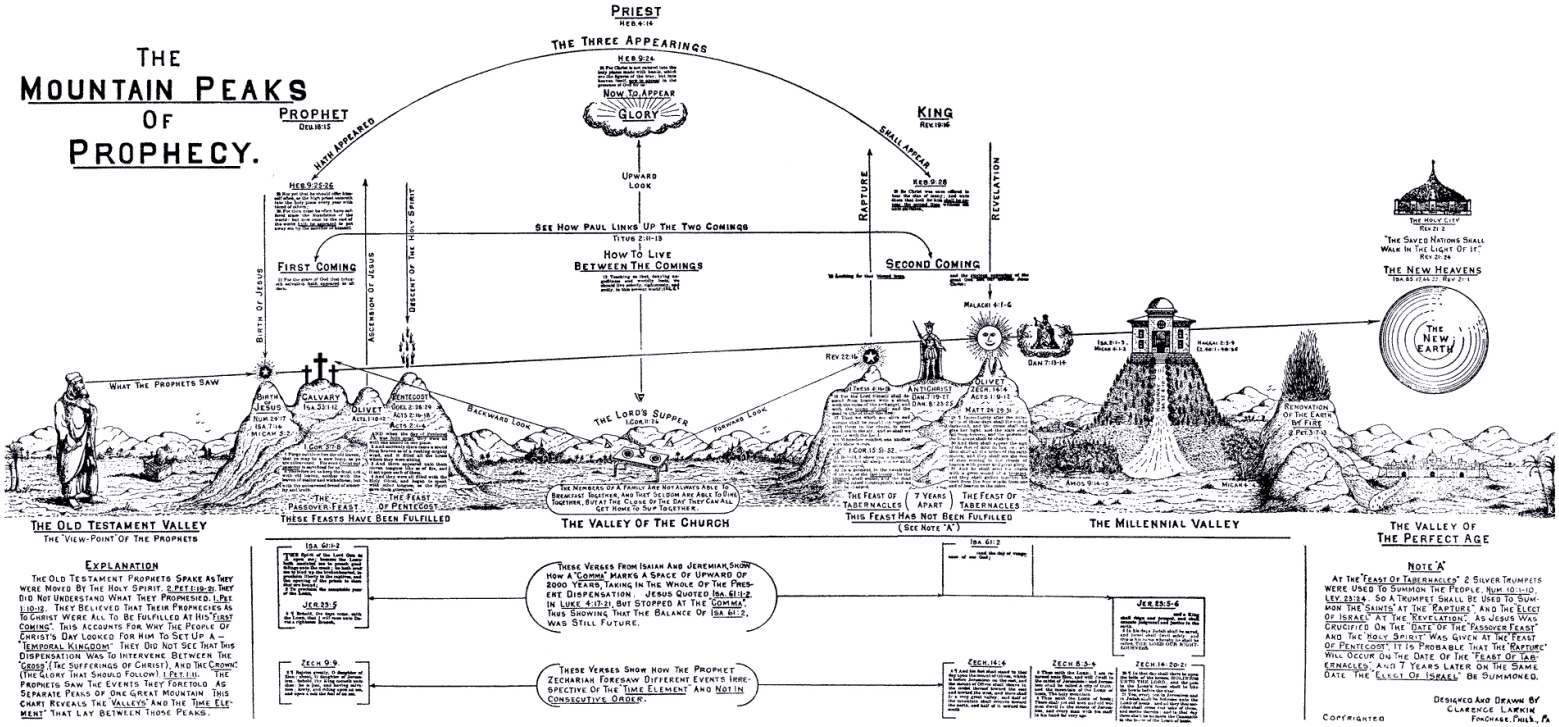


FIGURE 2.2: The interpretative Biblical infographic "The Mountain Peaks of Prophecy" (Larkin, 1918). *Image in the public domain.*

into a usable or communicable form (Card *et al.*, 1999). Interactions with information visualizations are also a form of *epistemic action* — actions that do not directly lead to a goal but result in a change in the person's world view to support a new way of thinking about a problem (Kirsh and Maglio, 1994). Neth and Payne (2002) refer to such actions as "thinking by doing". Epistemic actions are likely to play a role in sensemaking and knowledge crystallization tasks, cognitively separating interactive visualization from static information graphics. Yi *et al.* (2007) provide a taxonomy to organize many varieties of interactions used in InfoVis, framed around *analyst intent*. We also discuss interaction below as part of the *information visualization pipeline* (see §2.3.2).

While accepting the concept of external cognition, Scaife and Rogers (1996) review a number of studies and caution against blanket assumptions such as "color is better than black and white" or "interactive graphics are better than non-interactive graphics". They draw attention to the lack of solid research results on *how* interactive visualizations work as external cognitive aids (the diagrams studied by Larkin and Simon (1987) were static), and caution against generalizing that any particular information graphic or visualization may be useful as an external cognitive aid. Scaife and Rogers hypothesize that the success of a visual cognitive aid depends on three properties:

COMPUTATIONAL OFFLOADING The extent to which the external aid reduces effort.

RE-REPRESENTATION The way different external aids, with the same abstract structure, may make certain problems easier or more difficult to solve.

GRAPHICAL CONSTRAINING The way the graphical representation restricts or enforces the kinds of interpretations that can be made.

Since the cautions of Scaife and Rogers (1996) were published, it has been shown that interactive computer displays can augment human cognitive abilities to provide analytical power beyond what a computer or person is capable of alone (*e. g.*, Scott *et al.*, 2002; Wright *et al.*, 2000). Drawing on Bertin (1983), Larkin and Simon (1987), Tufte (1990), and others, Card *et al.* (1999) provide an enumeration of the ways in which InfoVis may amplify cognition:

- by increasing the memory and processing resources available to the users,
- by reducing the search for information,

- by using visual representations to enhance the detection of patterns,
- by enabling perceptual inference operations,
- by using perceptual attention mechanisms for monitoring,
- and by encoding information in a manipulable medium.

Licklider (1960) had a vision of "man-computer symbiosis": people and computers collaborating synergistically to accomplish tasks faster and with less effort. Many of the most successful examples of symbiosis, such as reduced search time and improved decision-making, come from graphics and visualizations which embody good design practices, such as using high data densities (Tufte, 2001). The next section will introduce the design influences which most strongly inform the case studies introduced in later chapters.

man-computer symbiosis

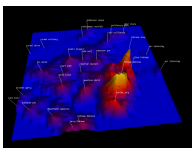
2.3 DESIGN INFLUENCES

Some of the best practices in InfoVis are adopted not only from experimental evidence, but from a wealth of well-justified theories and thoughtful design advice. The remainder of this section will explore the work of influential theorists of visual sensemaking processes (*e.g.*, Card *et al.*, 1999; Shneiderman, 1996), theorists of graphical perception (*e.g.*, Bertin, 1983), experimentalists in cognitive psychology (*e.g.*, Cleveland and McGill, 1984; Healey *et al.*, 1996; Ware, 2004), and experienced information graphic designers (*e.g.*, Stone, 2003; Tufte, 2006). We also discuss our own concept of *spatial rights* as a motivating design influence. Finally, this section introduces *heuristic approaches* to evaluation, which we use *prescriptively* to inform the design process.

2.3.1 *Visualization 'Mantra'*

Shneiderman (1996) offers a succinct model of a possible InfoVis process, called the "visual information-seeking mantra": "Overview first, zoom and filter, then details-on-demand". This four-step description of a typical visualization usage pattern can be read as a design guideline summarizing many of the commonalities of effective information visualization. Most importantly, it captures the need for visualizations to be effective on both a macro and micro level.

Zoom functionality allows the analyst to target a region of interest. Zooming on details can be geometric (for example, make regions of the display larger in order to read small words or differentiate between nearby nodes), or semantic (for example, provide more specific theme words or break clusters to show individual authors). *Filtering* can take several forms: (1) remove the context from the display, leaving only items of specific interest, (2) provide more detail on a focal region, abstract and display surrounding data (focus + context), or (3) show detail in a new window, highlight region of enlargement on the overview display (overview + detail).



Themescape (Wise et al., *1995)* © IEEE. Reprinted with permission.

We can examine this process using an imaginary scenario based on visualizing a collection of all the papers ever published at *IEEE InfoVis*. We walk through the process as visualized by Themescape (providing theme words mapped with a geographical metaphor) (Wise *et al.*, 1995) and Nodetrix (providing node-link and matrix diagrams linking co-authors) (Henry *et al.*, 2007), two visualizations which embody the 'mantra'.

Nodetrix (Henry et al., *2007)* © IEEE. Reprinted with permission.

The visualization first provides an *overview* of the entire data set, displaying high-level features of the data to allow the analyst to then specify a region of interest. For example, one may see major theme words organized in relation to one another in Themescape. Alternatively, the overview could represent the co-authorship networks at the level of research groups in NodeTrix.

Finally, *details-on-demand* provides more detailed features of the data. For example, by opening a list of definitions of a word when it is selected with the mouse, or revealing the publications list for an author of interest.

Beyond the 'mantra', Shneiderman includes three more important tasks that visualizations may support: *relate* — view relationships among items; *history* — keep a history of actions to support undo; and *extract* — allow extraction of sub-collections and of the query parameters.

Shneiderman's set of tasks is just one of many explanatory models of how an analytical process can proceed with interactive visualization. In fact, Shneiderman advises the 'mantra' is intended as "descriptive and explanatory" rather than prescriptive (Card *et al.*, 1999). Craft and Cairns (2005) conducted a review of literature citing the 'mantra' as a design influence, or using it in an evaluative way. They found little empirical evidence supporting its utility as a generative mechanism for designing visualizations. They call for a new holistic design methodology which is

prescriptive, measurable, and step-wise. However, they do not present one, and to our knowledge one does not yet exist.

As van Welie *et al.* (2000) caution, guidelines such as the 'mantra' are often problematic: they can be difficult to interpret consistently and may conflict with other guidelines. Isenberg *et al.* (2008) present results of a study which suggest the ordered process of the 'mantra' is not necessarily the best practice. After observing people working with paper-based information graphics, they found people naturally switched amongst the processes of information analysis. For example, one may wish to use a document content or lexical relationship visualization in the reverse order from the 'mantra', starting at a word of interest and interactively broadening the view to understand the context and relationships. Due to these cautionary results, in this work the 'mantra' is taken as an initial inspiration for an interaction process, not as the only valid (or always appropriate) way to explore a visualization.

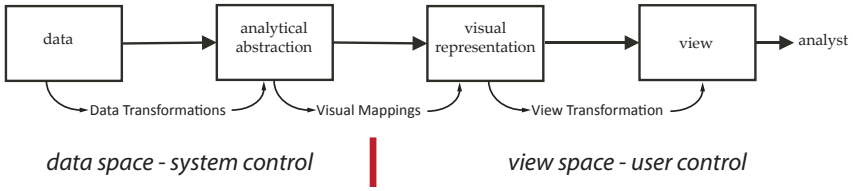
2.3.2 *The Information Visualization Pipeline*

Chi and Riedl (1998) proposed a model of visualization use composed of a series of steps, arranged as a pipeline, which describe the data transformations performed by the system and the visual transformations resulting from analyst interaction with a visualization (see Figure 2.3a). Card *et al.* (1999) present the model without a separation between system and analyst control. We adopt this modification — depending on the task, the analyst could be involved with data transformations (*i. e.*, the analyst controls the system). This model was further extended by Carpendale (1999) to include the presentation space. Additionally, we add the interaction possibility of access/edit/annotate data, to allow for loading and editing of source data.

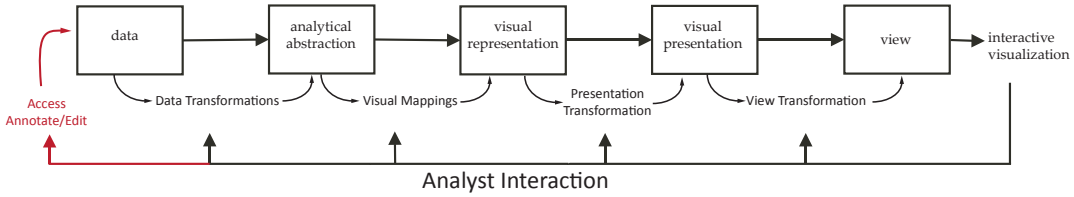
At each stage the analyst may provide feedback which can affect the data processing, visual encoding, presentation, or view. We illustrate the extended model of Carpendale (1999) in Figure 2.3b and discuss each part below.

In the pipeline there are five ways in which data is encoded (the boxes in Figure 2.3):

DATA The raw form of the data, such as a speech waveform, a plain text file, or database.



(a) The information visualization pipeline of Chi and Riedl (1998), including a division between system and user control.



(b) The information visualization pipeline of Carpendale (1999), including a presentation step, extended to allow an analyst to access/change the data.

FIGURE 2.3: The information visualization pipeline.

ANALYTICAL ABSTRACTION A processed form of the data, such as term vectors for text, or a list of co-citations collected from court cases.

VISUAL REPRESENTATION A basic visual form of the data, such as a tag cloud or graph.

VISUAL PRESENTATION An adjusted form of the representation, without changing the information content, for example, items may be highlighted or enlarged.

VIEW The final visible form, the window on the presentation space that is rendered to the screen.

Our version of the pipeline has five points at which the analyst may interact (the upward pointing arrows in Figure 2.3):

ACCESS/ANNOTATE/EDIT Accessing (viewing) the unmodified source data, or changing (editing/annotating) the source data.

DATA TRANSFORMATIONS Adjustments to how the data is transformed into an analytical abstraction, for example by applying a different keyword extraction method, or extracting additional metadata from a translation model.

VISUAL MAPPINGS Changes to the form of the visual representation, for example using a node-link graph instead of a space-filling graph.

PRESENTATION TRANSFORMATION Changes to the presented visual representation that do not change the information content or relations

between data items, for example, selecting new focus items to highlight, or rearranging nodes in a graph (without modifying links).

VIEW TRANSFORMATION Changes to way the presentation space is displayed, for example, rotation, pan, and geometric zoom.

Note that we make a distinction for viewing the source data, because while visual abstractions of textual data may be useful analytical tools, they are not replacements for reading. The absence of facilities to read source text was a primary complaint raised in a recent study of linguistic visualization (Wattenberg and Viégas, 2008). Viewing the source text can be seen as a type of details-on-demand (see §2.3.1). In our pipeline model, we define viewing unmodified source data as "access" data, but it could also be seen as a type of (minimal) flow through the pipeline, from stored bits to alphabetic characters to rendered text which may have presentation operations (highlighting) or view operations (scrolling) applied.

This model is often used to describe the way in which raw data becomes a final on-screen view. It can also be used as a design guideline showing the common transformations an interactive visualization may provide, as well as access points for analyst interaction. We introduce this model as we are interested in how the *depth-of-interaction* fits into the ecology of linguistic visualization (see §3.5).

2.3.3 *Visual Variables*

A selection of graphical dimensions were originally proposed by Bertin (1983, English Translation of "Sémiologie Graphique", 1967.) as retinal or visual variables, because they have to do with features of the visual space to which the human retina is particularly sensitive. These variables are ways to visually encode information upon the elementary marks of graphics: points, lines, areas, surfaces, and volumes.

Bertin's original visual variables are outlined in Figure 2.4. Carpendale (2003) provides an updated analysis of visual variables, adding motion, colour saturation, texture, and transparency, which are especially relevant to interactive visualization. Each visual variable can be characterized by five features:

SELECTIVE Can a change in this variable make it easier to select the changed mark among other unchanged marks?

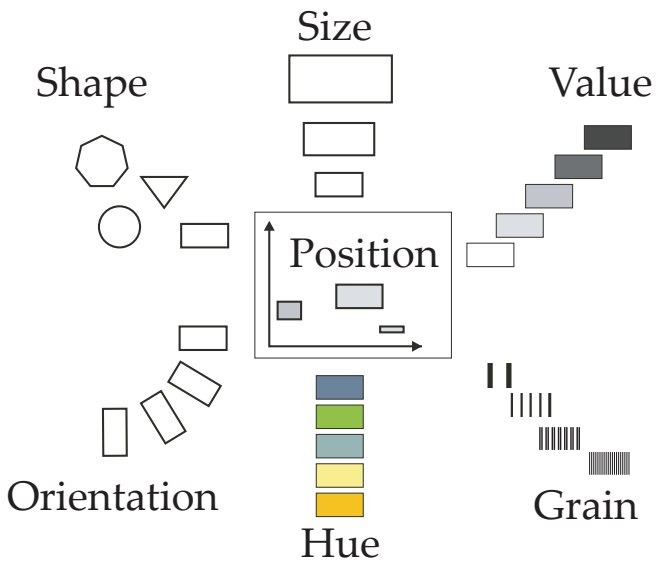


FIGURE 2.4: Some visual dimensions useful for encoding data, adapted from (Bertin, 1983).

ASSOCIATIVE Can marks that have like values of this variable be grouped in the presence of other marks with different values?

QUANTITATIVE Can a quantitative value be read from this variable?

ORDER Do changes in this variable support ordering of data?

LENGTH How many values of this variable can be associated or distinguished without losing its functionality?

Probable values of these features for each variable have been suggested by Bertin (1983) and Carpendale (2003), and are summarized in Table 2.1. These characteristics help a designer select the appropriate variable to encode the dimensions of the data to be visualized. For example, if the number of words per document is an important feature of a corpus to be visualized, it would be inappropriate to encode the number of words using shape or colour, as these are not quantitative variables. Characteristics of visual variables can also be used to evaluate visualizations and diagnose usability problems.

Visual variables have been further investigated by Cleveland and McGill (1984, 1985) in a series of experiments comparing human accuracy at carrying out different *elementary perceptual tasks* (*i. e.*, reading a value of particular visual variable) for various data types. The results of these studies generally agree with the categorizations of Table 2.1, but also provide

Variable	*Select*	*Associative*	*Quantitative*	*Order*	*Length*
Position	Yes	Yes	Yes	Yes	Variable
Size	Yes	Yes	Approx.	Yes	Assoc: 5; Dist: 20
Shape	With Effort	With Effort	No	No	Very Large
Value	Yes	Yes	No	Yes	Assoc: 7; Dist: 10
Hue	Yes	Yes	No	No	Assoc: 7; Dist: 10
Orientation	Yes	Yes	No	No	4
Grain	Yes	Yes	No	No	5
Texture	Yes	Yes	No	No	Very Large
Motion	Yes	Yes	No	Yes	Unknown

TABLE 2.1: Properties of visual variables (Bertin, 1983), extended by Carpendale (2003). Length abbreviations: *Assoc.* – associate, *Dist.* – distinguish.

a rank order of variables for different data types: position, followed by 1D size (segment length) is most effective at conveying quantitative values, while colour and shape can be accurately read for categorical information. A particularly interesting finding was that the amount and direction of estimation error depends on the true value; for example, people can accurately estimate the value of small angles while mid-range angles are problematic.

Guidelines for the selection of appropriate colours for use in InfoVis have been suggested by Brewer and Harrower (2002) and Stone (2003). They provide useful advice on topics such as selecting colours for highlighting, creating colour maps for various sorts of scaling, and avoiding alienating viewers with colour-blindness.

Rapid Processing

There are some characteristics of visual scenes that are identified rapidly (under 200 ms) and accurately in human vision (Healey *et al.*, 1996). When these visual properties were first catalogued, they were called *preattentive properties* and placed in opposition to other properties that were thought to be detectable only with active attention. Figure 2.5, top row, illustrates a search task for which hue is a 'preattentive' feature. Recognizing the types of visual variables that are rapidly processed is important to InfoVis design because these properties can be used to make important features noticeable at a glance.

There is disagreement about the mechanism of this phenomenon. The notion that some features are recognized without the need for focused

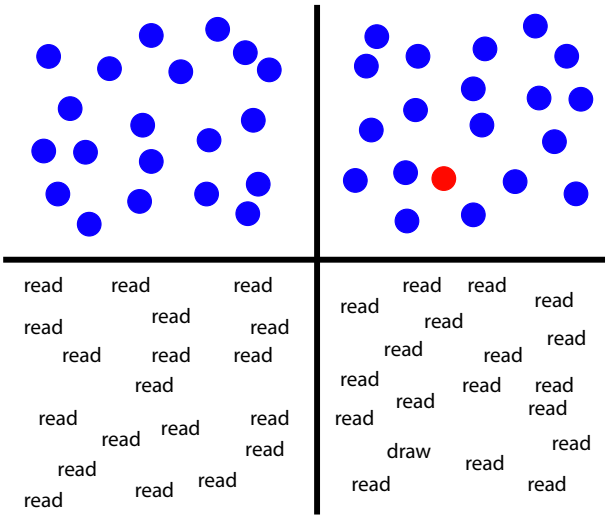


FIGURE 2.5: Hue is a 'preattentive' visual feature. Viewers quickly can tell if the target red circle is absent (top-left) or present (top-right) in the presence of distractors. However, text is not 'preattentive', requiring sequential search to discover a target word ('draw', present in the bottom-right).

Form	*Colour*	*Motion*	*Spatial*
orientation	hue	flicker	position
length	value (intensity)	direction	steroscopic depth
width	lighting direction		convex/concave
collinearity			grouping
size			
curvature			
blur			
added marks			
shape			
number			

TABLE 2.2: 2D visual features that can be perceived in under 200ms. Compiled from Healey *et al.* (1996) and Ware (2004, pp. 151–152). For further details on each feature, the reader is referred to the list of original sources found in those references.

attention has fallen out of favour, and some of the previously claimed preattentive properties have been found not to be so (*e.g.*, Wolfe, 2003; Wolfe *et al.*, 2003). Indeed, there are ongoing debates about the exact nature of many cognitive processes, including language learning (Pinker, 1997). This dissertation is not concerned with the *internal processes*, but rather how empirical evidence can inform design.

Despite the debate about the underlying cognitive processes, it remains a fact that the speed at which we can recognize changes in certain visual properties remains nearly constant as the scale of data increases (the time to find a red spot among 100 or 1000 blue spots is essentially equal). As a rule of thumb, Ware (2004, p. 151) suggests if a feature is recognized at a rate faster than 10ms per distractor, it can be considered 'preattentive'. Search time for other visual features scale linearly at 40ms per distractor or more (Treisman and Gormican, 1988). Table 2.2 presents a summary of the visual features that can be rapidly processed. In our designs, we endeavour to use visual properties such as hue, value, and blur variation to provide visual *pop-out* to items of importance or focus. However, despite the preattentive nature of shape, finding targets in text representations does not exhibit the speed profile of the so-called preattentive visual features (Figure 2.5, bottom), likely because text combines complex shapes and sizes, curvatures, and widths. The effects of interference when two or more of these properties are used in conjunction is a matter of ongoing research, with findings suggesting that in most cases the 'preattentive' speeds are lost (Healey, 2007).

Reflecting on Table 2.1, Ware (2004, p. 182) suggests that the length of a visual variable is decreased when 'preattentive' processing is desired. Distinguishable hues reduce to 8, orientation to 4, size to 4, and all other variables are also in the single-digit range.

2.3.4 *Gestalt Perception*

The formation of internal models through perception can be described with *Gestalt theory*. Founded by Max Wertheimer in 1912, this branch of psychology is based on a belief that humans often perceive more than what our physical senses receive as input (*Gestalt* means *pattern* in German). We fill in the gaps of perception using our internal models. The heightened perception is described by laws which describe human tendencies to make connections between disconnected objects. Though

Gestalt Laws	*Explanation*
Similarity	Similar objects (in size, shape, colour, *etc.*) tend to be grouped together.
Proximity	Objects near one another are seen as a unit.
Continuity and Connectedness	Smooth and continuous lines will be perceived before discontinuous ones. Objects are perceived as a group if connected by smooth or straight continuous lines.
Closure	Contours with gaps will be perceptually closed. Regions are perceived as having an "inside" and an "outside". In windowed visualizations, closed contours are used to group objects and segment display space.
Symmetry	Symmetrical pairs are perceived more strongly than parallel pairs. Scenes are perceived as being made up of symmetrical parts if possible.
Relative Size	Smaller regions in a pattern are perceived as objects, larger regions as the background.
Common Fate	Objects moving in the same direction appear as a unit.
Familiarity	Elements are more likely to form a group if the group forms a familiar shape or appears meaningful.

TABLE 2.3: Gestalt laws important for InfoVis design (Card *et al.*, 1999; Ware, 2004).

much of the theory of Gestalt psychology is controversial (King and Wertheimer, 2005), the basic laws of perception offer some guidance for information visualization. The Gestalt laws are presented by Ware (2004) as design principles and are summarized in Table 2.3. Additional perceptual studies reported by Ware highlight potential additional Gestalt laws: perceived groupings based on transparency, perceived direction of movement based on vector illustration, and perceived contours from arrangement of shapes. Information visualization designers should be aware of these and the basic Gestalt laws in order to use them to their advantage and avoid the inadvertent visual perceptions that can occur, as in optical illusions.

2.3.5 *Clear and Concise Design*

The works of Tufte (1990, 2001, 2006) fall into the "design advice" category. His recommendations, such as avoiding uninformative elements (*chart junk*), avoiding misleading visual encodings (*lie factors*), and maximizing the data-ink to non-data-ink ratio, are derived from his own years of experience analyzing and creating information graphics. Generally helpful, we strive to follow them in this work. One particularly relevant

contribution is his recommendation of improving *usability* and *learnability* by visualizing with *small multiples*: repeated, small visualizations which as a group encode a lot of information, but individually are easy to read. Once an observer knows how to read one of the multiples, they can read all of them. We use this technique in Chapters 4, 6, and 7.

2.3.6 *Spatial Rights*

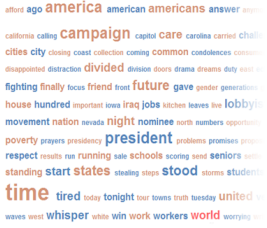
The research we have reviewed indicates that the spatial positioning of data items is the most visually powerful of the available visual variables. Spatial position (2D) is selective, associative, quantitative, 'preattentive', and ordered. It also has a variable (but long) length and is involved in our perception of grouping through Gestalt perception (grouping is also 'preattentive').

Part of the process of choosing or designing a representation is called the *spatialization*, or layout, of the data. When data is spatialized based on a particular data dimension or relation, we say that data dimension or relation has been granted *spatial rights*, to indicate the visual primacy that data dimension or relation will hold (Collins and Carpendale, 2007).

SPATIAL RIGHTS The assignment of position as a visual encoding for a data dimension or relation.

As a simple linguistic example, consider the spatialization of words in a tag cloud. A tag cloud is a spatialized grid of words, sized, and perhaps coloured, to represent aspects of those words, such as frequency of use in a document, or recency of use in a discussion. The layout is usually packed in rows, so the spatialization choice equates to choosing how to order the words.

If the most important task to support was visual search for the presence of a word of interest, an alphabetical layout would be best. In this case, the alphabetic dimension of the data would have spatial rights, while other aspects that may be of interest, such as frequency, would be encoded with other visual variables. Assigning alternating colour to adjacent words to distinguish them from one another, and encoding frequency as size would result in a tag cloud similar to the *Many Eyes* tag cloud (see margin figure). While size is also a powerful visual encoding, judging relative size, especially of non-adjacent words, would be difficult to do accurately (Larkin and Simon, 1987). In this way frequency and frequency rank become



Many-Eyes Tag Cloud (Viégas et al., *2007)*

visually secondary to alphabetical order through the assignment of spatial rights.

We can imagine an alternative situation in which the rank of frequencies was the most important factor, but recency information was also desired. In this case, we would choose to position the words in order by rank (assign spatial rights to rank). This would permit an easy reading of rank order. We could encode the precise frequency value to word size, since size is approximately quantitative, and we could assign recency to the colour value, as value is ordered. In this case, serial search of the whole visualization would be needed to determine the presence or absence of a word of interest.

We will recall the discussion of spatial rights later in the dissertation, presenting in Chapter 7 a method for rendering a set relation over an existing layout without disrupting the primary relation's spatial rights. In Chapter 8 we formalize ways to assign spatial rights in multi-relation visualizations and introduce a technique to allow for the re-use of the spatial dimension in the same view (multiple spatial rights).

2.3.7 *Heuristic Approaches*

Much of the foundational knowledge described in previous sections has been reiterated in a set of heuristics for information visualization evaluation compiled by Zuk *et al.* (2006). Heuristic evaluation is a well known technique in human-computer interaction research, but is not widely used in InfoVis. Heuristic evaluation consists of a series of tests for specific problems, carried out by a set of evaluators. For example, one may use "Ensure luminance difference between red and green." as a heuristic to avoid creating designs which cannot be used by people with red-green colour-blindness. As described by Zuk *et al.*, heuristics such as these offer cheap, fast, and easy evaluation, appropriate for application during iterations of design. They can also be read prescriptively, as design guidelines. They heuristic sets proposed by Zuk *et al.* (2006) draw on the work of Amar and Stasko (2004), Shneiderman (1996), and Zuk and Carpendale (2006). We select the final subset, "perceptual and cognitive heuristics," which were determined to be the most helpful and easy to apply. These are listed in Table 2.4 and will be treated as general guidelines when discussing design decisions in subsequent design study chapters.

Heuristic
Ensure visual variable has sufficient length
Don't expect a reading order from colour
Colour perception varies with size of coloured items
Local contrast affects colour and grey perception
Preattentive benefits increase with field of view
Quantitative assessment requires position or size variation
Preserve data to graphic dimensionality
Put the most data in the least space
Remove the extraneous detail
Consider Gestalt Laws
Provide multiple levels of detail
Integrate text wherever relevant

TABLE 2.4: Heuristics for evaluation of perceptual and cognitive factors in InfoVis design (Zuk *et al.*, 2006).

2.4 A METHODOLOGICAL NOTE ON EVALUATION

There are a number of challenges facing the evaluation of InfoVis in general. For an overview of the challenges of evaluating InfoVis, Plaisant (2004) provides a comprehensive account. Numerous techniques have been suggested, including lab-based user studies (Kosara *et al.*, 2003), long-term deployment case studies (Shneiderman and Plaisant, 2006), insight-based experiments (Saraiya *et al.*, 2005), automated evaluation of perceptual organization (Wattenberg and Fisher, 2004), visual quality metrics (Bertini and Santucci, 2006), measuring extrinsic effects such as worker productivity (McNee and Arnette, 2008), and using heuristic evaluation (Zuk and Carpendale, 2006).

There is no agreement on when it is appropriate to use quantitative techniques such as user studies. However, Kosara *et al.* (2003) advise that user studies should be reserved for evaluating specific techniques, especially in a comparative sense. Quantitative studies are also appropriate and informative to understand human perception. Questions such as, "which colour scale produces more accurate readings?", and "at what blur factor is a separation between sharp and fuzzy distinguishable?" are possible to answer with a well-designed perceptual study. This falls outside the field of InfoVis and is more appropriately studied by cognitive scientists. The previously discussed work of experimentalists such as Cleveland and McGill (1984) fall into this category. We plan to conduct a study of

this sort on Bubble Sets (see Chapter 7) in collaboration with perceptual psychologists as future work.

Greenberg and Buxton (2008) caution against using usability studies to evaluate early stage, exploratory research. Getting bogged down on the types of details measurable with user studies when inventing prototypes that address new problem areas can eliminate ideas too early. Also, when there are few or no compartors to compare against (which is often the case in InfoVis), it is challenging to create a study that convincingly says anything about a holistic system. Such studies are often guided to prove the new technique is "useful for at least one thing" (Greenberg and Buxton, 2008), and are therefore inherently biased and not very useful.

Formal evaluation is only one type of validation of an information visualization. Visualizations can be considered valuable contributions based on many other metrics: Does it render faster than previous systems? Is it algorithmically more efficient (reduced complexity)? Does it reveal information that was previously not visible at all? Does it introduce a new, generalizable technique to the community? Does it address a space in a formal framework of research where previously there was nothing? Within each of our design studies, we address these questions where an answer can be given with confidence.

As the design studies presented in this dissertation all fall into the category of *early prototype*, and have few comparators (and fewer, perhaps none, with published evaluations), we made the choice to forego formal evaluation (studies) in favour of a broader exploration of the space of linguistic visualization. Additionally, while they would be an interesting follow-up to this research, in order to explore a breadth of problem areas within the *space of linguistic visualization*, we did not conduct long-term case study evaluations. Within each design studies, we instead investigate and discuss a range of design options in the light of the aforementioned theories and guidelines, and demonstrate through example the types of information one may see with a visualization that was not previously available (at least in a practical sense). It may be informative in future to apply heuristic evalution to the design studies, using a comprehensive set of heuristics for InfoVis as suggested by Zuk (2008).

2.5 SUMMARY

Information visualizations, when well-designed, have the potential to act as external cognitive aids. The types of tasks a particular visualization will support depends on the representation used, and the interaction provided. There are a number ways in which design can be guided to increase the chances that a particular visualization will be useful and usable, including well-established design guidelines based on theories of sense-making and visual information-seeking, and experimental evidence about the capacities of human visual perception.

Information visualization techniques have been applied to a broad range of linguistic data, with varying sophistication in the underlying natural language processing (NLP) (the *data and analytical abstraction*) and the method of visualization (the *respresentation, presentation, and view*). In the following chapter we will define an informal framework for mapping the space of linguistic visualization, and in the process examine examples of work of varying linguistic sophistication.

3

THE SPACE OF INTERACTIVE LINGUISTIC VISUALIZATION

Distant reading . . . is not an obstacle, but a specific form of knowledge: fewer elements, hence a sharper sense of their overall interconnection. Shapes, relations, structures. Forms. Models.

— Franco Moretti (2005, p. 1)

What classifies as 'linguistic visualization'? Luhn (1960) introduced the keyword-in-context (KWIC) technique for aligning instances of a word of interest in a table, alongside left and right context words (Figure 3.1). This relatively simple method of extracting data and representing it in an organized fashion was immensely helpful to subsequent researchers (*e. g.*, Schreibman *et al.*, 2004; Wattenberg and Viégas, 2008). Although the creation of a KWIC display is clearly an instantiation of the information visualization pipeline (§2.3.2), it is rarely described as a *visualization*.

From Luhn's early work in representing linguistic data to amplify cognition, we have come a long way. There are now annual sessions on *text visualization* at visualization conferences, and conferences such as the *Annual Meeting of the Association for Computational Linguistics* (ACL) have begun to warm to visualization as a technique worth exploring. Although interactive visualization contributions have generally appeared as posters, not papers, at ACL (*e. g.*, DeNeefe *et al.*, 2005; Smith and Jahr, 2000), things may be changing. The recent large attendance and enthusiastic feedback a tutorial on information visualization (InfoVis) and computational linguistics (CL) may be a positive sign of things to come (Collins *et al.*,

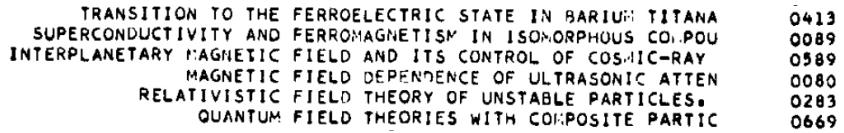


FIGURE 3.1: An excerpt of Luhn's (1960) original keyword-in-context index for technical literature. The different uses of the world 'field' can be seen. *Reprinted with permission.*

2008). Additionally, many people outside research, such as artists, designers, and bloggers have become actively engaged in visualizing language. With this diverse history, and recent explosive growth, it has become clear we need a descriptive framework to understand the varied goals of and approaches falling under the umbrella *linguistic visualization*: "The presentation of linguistic data through visual representations designed to amplify cognition or communicate linguistic information.".

The types of linguistic data forms include text, augmented text (text associated with images), speech, gestural languages, and singing (speech with music). While visualizations of speech (*e. g.*, Levin and Lieberman, 2004) and gesture (Falletto *et al.*, 2009) have been reported, this dissertation focuses only on visualizations of text, augmented text, and speech provided as text from an automatic speech recognition (ASR) system.

In order to understand the rich space of past work in linguistic visualization, we contribute a descriptive framework which categorically partitions the literature based on human- and application-centred categories. This informal structure is emergent from the collection of related work and represents the types of questions we, as researchers, ask when designing or reading about examples of linguistic visualization. The dimensions we propose are listed briefly in Table 3.1. While we suggest four useful ways of dividing the space, each dimension alone can be used to classify most examples of linguistic visualization. Therefore, we will use the first of each of the human- and application-centred categories (Community of Practice, Problem Area) to present examples in depth, and briefly overview the other dimensions.

Human	COMMUNITY OF PRACTICE	Who created the visualization?
	TARGET AUDIENCE	Who is the visualization intended for?
Application	PROBLEM AREA	What problems is the visualization intended to solve?
	LEVEL OF INTERACTIVITY	How can a person interact with data/representation/view?
	TYPE OF LINGUISTIC RESOURCES	What linguistic resources and algorithms drive the visualization?

TABLE 3.1: The dimensions of the *space of linguistic visualization*.

As we alluded to in Chapter 1, the examples of linguistic visualization range from highly interactive interfaces with low linguistic sophistication, to linguistically profound, but static, information graphics. For some insight into this divide, we first examine selected examples from the varied communities which have been creating linguistic visualizations. The remaining sections of this chapter will examine the other dimensions. At the beginning of each section, we introduce an overview figure of the collection of reference works discussed in this dissertation, grouped into sets according to the dimension being discussed. The related work introduced in this chapter represents 2 or 3 canonical examples for each value along each dimension. Recent examples were preferred when available; the collection of related work is representative of the space of linguistic visualization, but is not intended to be exhaustive. The Bubble Sets visualization technique used to illustrate the relationships amongst related work is introduced in detail in Chapter 7.

3.1 TERMINOLOGY OF LINGUISTIC COMPUTING

While there is significant overlap between the fields of computational linguistics (CL) and natural language processing (NLP), within this dissertation we will adopt the view that they are different. The *Association for Computational Linguistics* defines CL as: "The scientific study of language from a computational perspective. Computational linguists are interested in providing computational models of various kinds of linguistic phenomena" (Sproat, 2005). They go on to describe that work in CL may be motivated by either a drive to model or explain an observed linguistic or psycholinguistic phenomenon or by a technological need for a working component of a computer-driven natural language system.

However, the authors of foundational textbooks in the field, such as Jurafsky and Martin (2008) and Manning and Schütze (1999) draw a distinction between these two approaches. The first, modelling of linguistic or psycholinguistic phenomena (*e. g.*, how people learn language, or how we understand and process idioms) is CL, while the development of algorithms for applied purposes, such as summarization, translation, and speech recognition, falls under NLP. Sometimes speech processing is considered a third area, but we will include it in NLP for our purposes. NLP is less concerned with grounding in cognitive linguistic principles and theory, and more concerned with achieving accuracy, precision, and

generally satisfying and convincing results. 'Natural' language here refers to fully expressive human languages such as English or Cantonese as opposed to formal languages such as the programming languages Java or C++.

The design studies outlined in this dissertation are generally examples of connecting InfoVis with NLP, although some of the resources used, such as *WordNet* (see Chapter 5) have been developed by the CL community. The specific contributions, especially VisLink (Chapter 8), may be broadly applicable to NLP, CL, and InfoVis. Many of our works, being based on underlying data structures of NLP systems, externalize the ways in which linguistic experts conceptualize language structure for the purposes of NLP. We do not claim these are representations of the ways language is structured *in our minds* — but that would be an interesting area for future research.

3.2 COMMUNITY OF PRACTICE

Within our broad definition of *linguistic visualization*, there are several communities approaching the problem from different directions. Our approach is grounded in NLP and InfoVis, but recognizing contributions from the wider community is informative. Figure 3.2 provides an overview of the related work in linguistic visualization, grouped according to community of practice.

3.2.1 *Computational Linguistics*

The computational linguistics community is interested in computational techniques to increase our understanding of language and linguistic processes. It publishes many static information graphics to communicate research results and to explain theories (*e.g.*, Figure 3.3a), but rarely does it publish interactive information visualizations. Exceptions include a method to map overlaps between semantic concepts across languages (Ploux and Ji, 2003) (see Figure 3.3b) and visualizations created to provide access to the results of linguistics research, such as the *Visual Walpiri Dictionary* (Manning *et al.*, 2001).

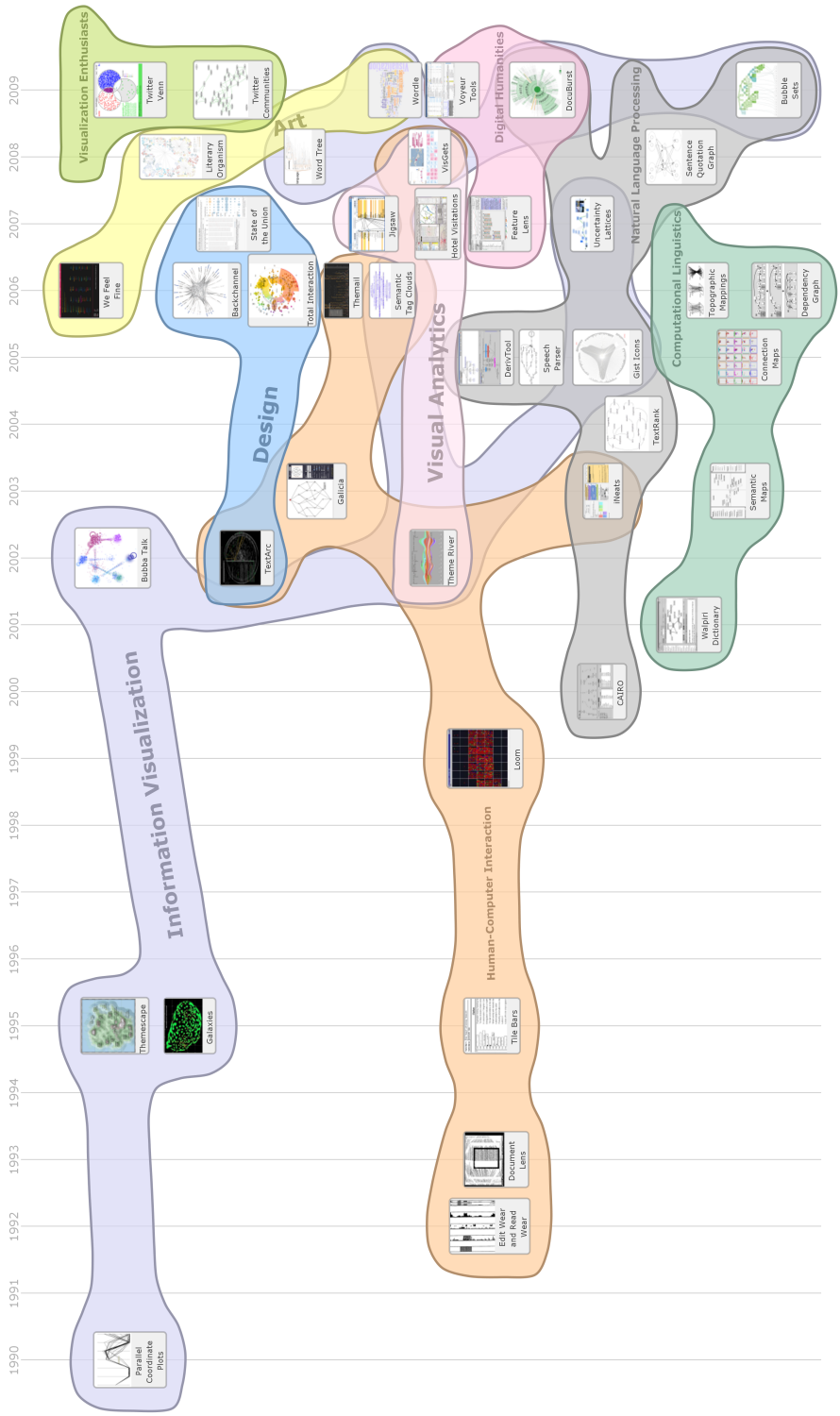
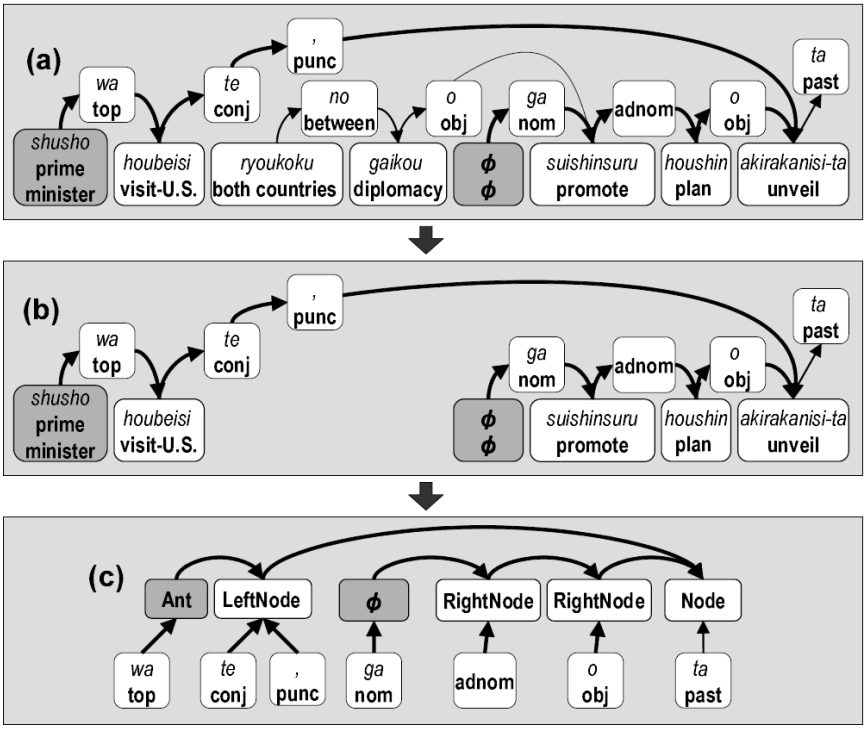
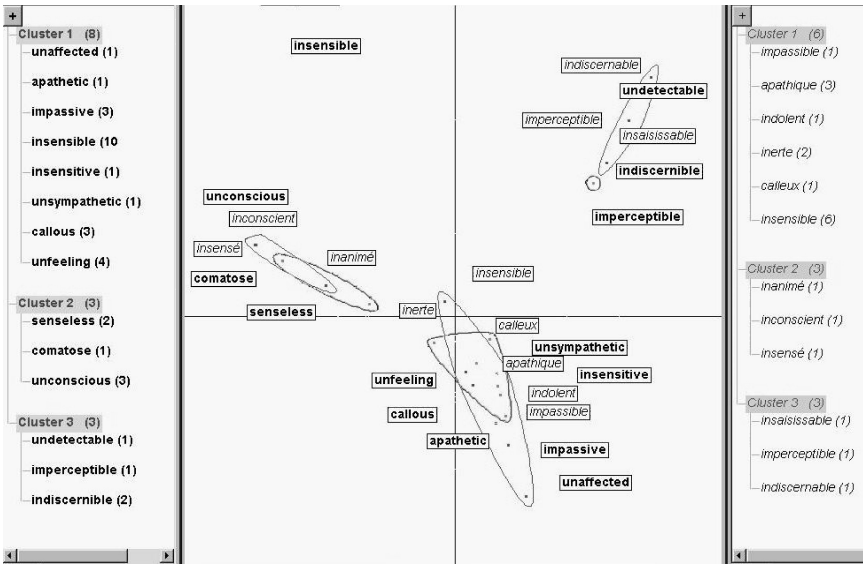


FIGURE 3.2: A Bubble Sets timeline illustrating the related work grouped by community of practice.



(a) Static information graphic representing the path between a zero-pronoun and its antecedent (Iida *et al.*, 2006).



(b) Interactive interface for exploring overlapping semantic maps (spatializations of words representing a concept) between English and French (Ploux and Ji, 2003).

FIGURE 3.3: Linguistic visualizations from the computational linguistics community. *Reprinted with permission.*

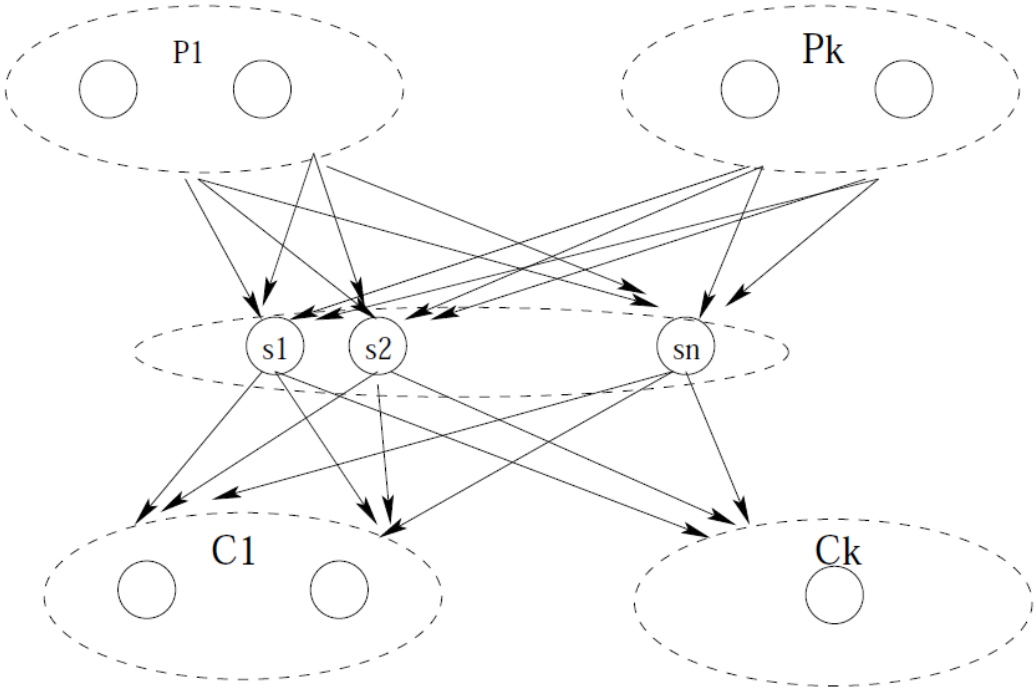
3.2.2 *Natural Language Processing*

Closely related to CL, researchers and engineers in the natural language processing community are interested in creating ever better and faster algorithms to process and transform linguistic data (*e. g.*, summarization, translation, spelling correction). This community also uses information graphics to disseminate results of research (*e. g.*, Figure 3.4a). A notable interactive contribution is the *iNeATS* interactive multi-document summarization interface (Leuski *et al.*, 2003), an example which bridges the linguistic visualization divide in its close coupling of summarization technology with interactive visualization of document content (see Figure 3.4b).

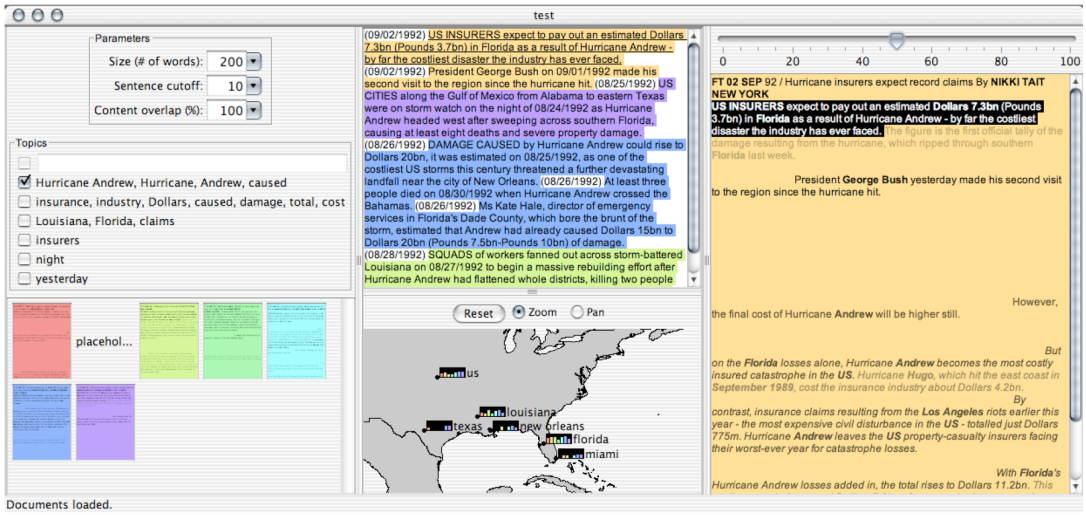
CAIRO (Smith and Jahr, 2000) and *DerivTool* (DeNeefe *et al.*, 2005) are two interactive visualizations for machine translation (MT) researchers to explore and refine their data models. *CAIRO* reveals bilingual word alignments and probabilities. *DerivTool* puts a human in the driver's seat, presenting the researcher with the set of choices considered by the algorithm at each step of processing (see Figure 3.5). We will return to the topic of statistical MT visualization to assist research in Chapter 7. Beyond these two examples, there is little evidence of interactive visualizations being used *in the process of research* as a technique to help NLP researchers improve their algorithms and data models.

3.2.3 *Human-Computer Interaction*

Human-computer interaction (HCI) researchers have long been interested in developing interfaces and interaction techniques to assist reading, editing, and navigating documents using a computer (*e. g.*, Alexander *et al.*, 2009; Appert and Fekete, 2006; Guiard *et al.*, 2006; Hill *et al.*, 1992; Robertson and Mackinlay, 1993). This community is generally not concerned with the *content* of the linguistic data, but rather the task, the context and technology of deployment of the interface, and usability issues related to interacting with the data. For example, Hill *et al.* (1992) described *Edit Wear and Read Wear*, a method for graphically depicting the reading and editing history of a document through an augmented scroll bar (see Figure 3.6a). Robertson and Mackinlay (1993) presented a magnification lens for document reading which provided a details-in-context view to allow reading of text while getting a feeling for the overall layout of the



(a) Static information graphic describing the creation of a sentence quotation graph (Carenini *et al.*, 2008).



(b) The interactive *iNeATS* multi-document summary interface, an example of interactive visualization coupled with NLP, bridging the linguistic visualization divide (Leuski *et al.*, 2003).

FIGURE 3.4: Linguistic visualizations from the natural language processing community. *Reprinted with permission.*

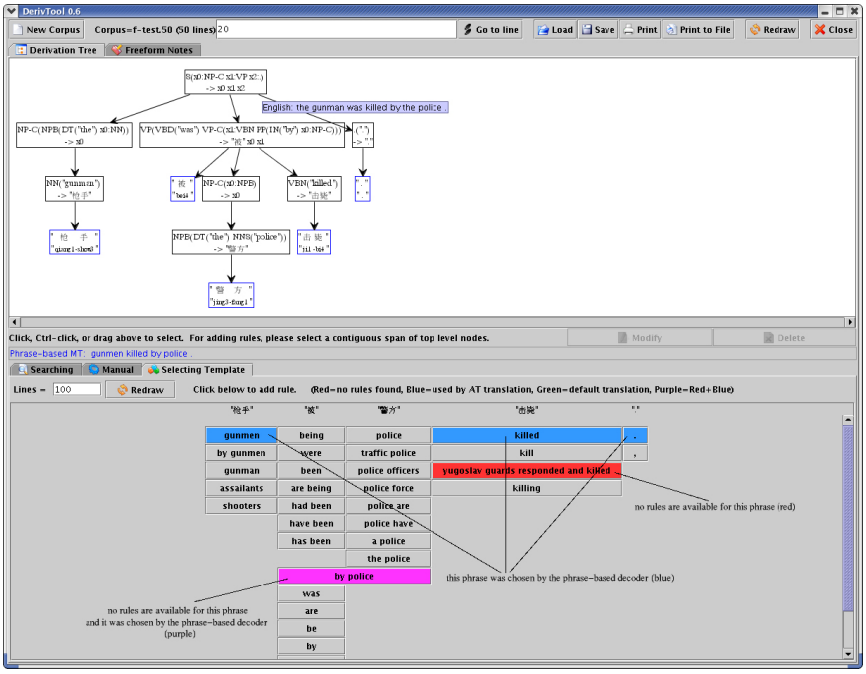
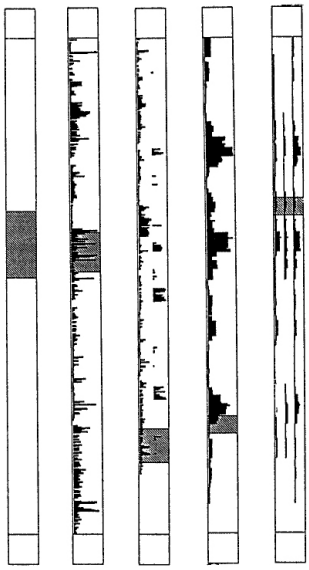


FIGURE 3.5: *DerivTool*: a method of interactively visualizing and directing the process of translating a sentence (DeNeefe *et al.*, 2005). *Reprinted with permission.*

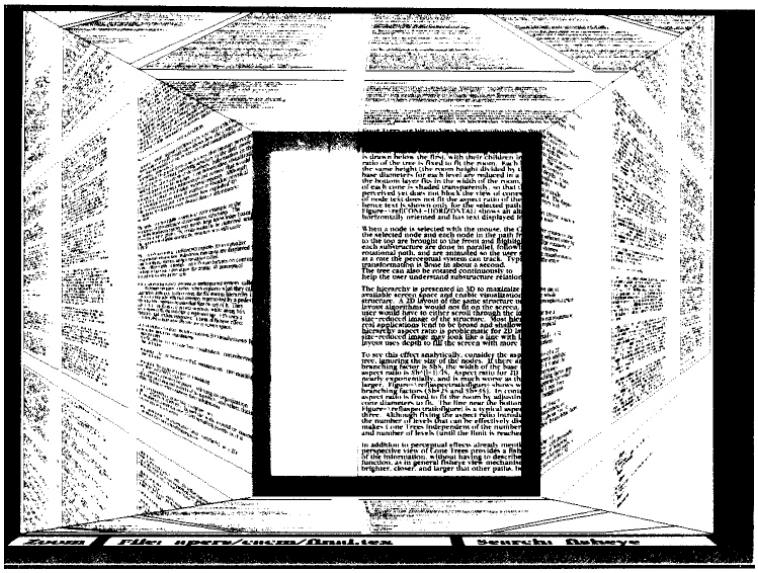
document (see Figure 3.6b). This community has also contributed foundational research which informs linguistic visualization, such as Grossman *et al.*'s (2007) investigation of the effects of orientation on text legibility in volumetric displays.

3.2.4 *Digital Humanities*

Digital humanities (DH), also known as humanities computing, is the field of research dedicated to using computing in the process of humanities research. Areas of inquiry range from literature and history to music and philosophy. Moretti (2005) argues that literary scholars should "stop reading books, and start counting, graphing, and mapping them instead". As indicated by the epigraph of this chapter, Moretti calls this approach *distant reading*. Computational efforts to facilitate distant reading, such as text analyses to count words and to map entity relations within a novel are examples of digital humanities research. Linguistic visualization research in this field is often undertaken in interdisciplinary teams, including do-



(a) *Read Wear* introduced an augmented scroll bar to reflect the readership history of a document. (Hill *et al.*, 1992).



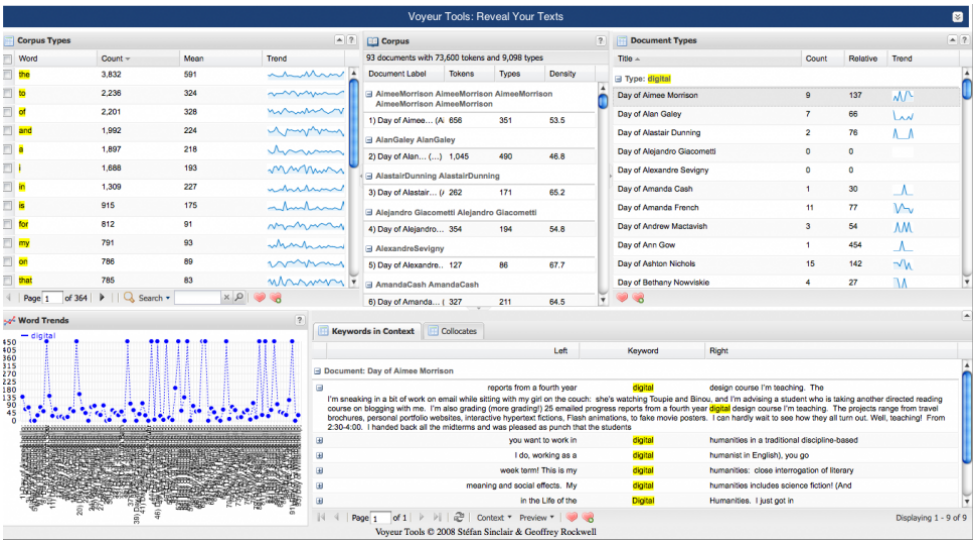
(b) *The Document Lens* provides a distortion view (presentation transformation) to allow legible rendering of a document while providing global context (Robertson and Mackinlay, 1993).

FIGURE 3.6: Linguistic visualizations from the human-computer interaction community. © *ACM, Inc. Reprinted with permission.*

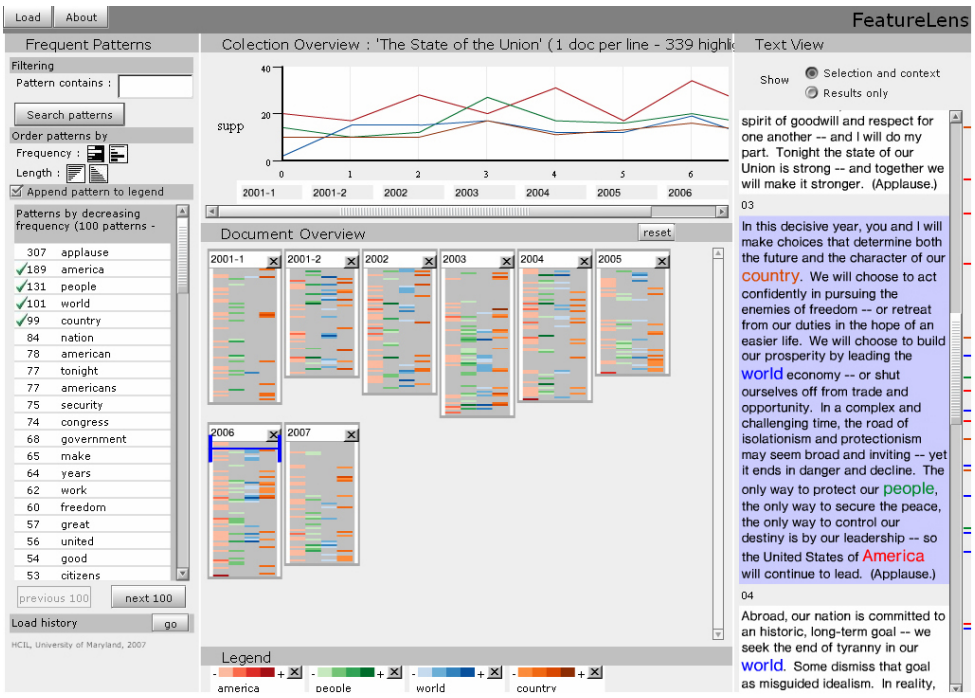
main experts (humanities scholars) and CL, NLP, and InfoVis researchers. A common goal is to create re-usable, general, online tools to support fellow humanities scholars in their research (Schreibman *et al.*, 2004).

The *Text Analysis Portal for Research* (TaPoR) is such an interdisciplinary digital humanities research consortium in Canada (`http://portal.tapor.ca`). Visualization contributions from this group include the *Voyeur Tools* (Sinclair and Rockwell, 2009) (see Figure 3.7a). *Voyeur Tools* visualize word frequencies, collocates, and trends in a corpus. *Voyeur Tools* use the representation concepts of sparklines (Tufte, 2006), small multiples (Tufte, 1990), and KWIC displays (Luhn, 1960). However, the text analysis is based solely on word counting.

The *Metadata Offer New Knowledge* project (MONK) in the United States (`http://www.monkproject.org`) also takes an interdisciplinary team approach to digital humanities research. They contribute the *Feature Lens* (Don *et al.*, 2007), which consists of an NLP component to count 3-grams (three-word sequences) and detect sets of co-occuring 3-grams. The NLP component is closely connected to an interactive visualization which illuminates



(a) *Voyeur Tools* (alpha-release) provides several simple, minimally interactive visualizations to reveal word frequencies and trends of use in a document collection (Sinclair and Rockwell, 2009). *Reprinted with permission.*



(b) *The Feature Lens* combines a text analysis algorithm for detecting multi-word patterns with a visualization for examining detected patterns in detail (Don *et al.*, 2007). © *ACM, Inc. Reprinted with permission.*

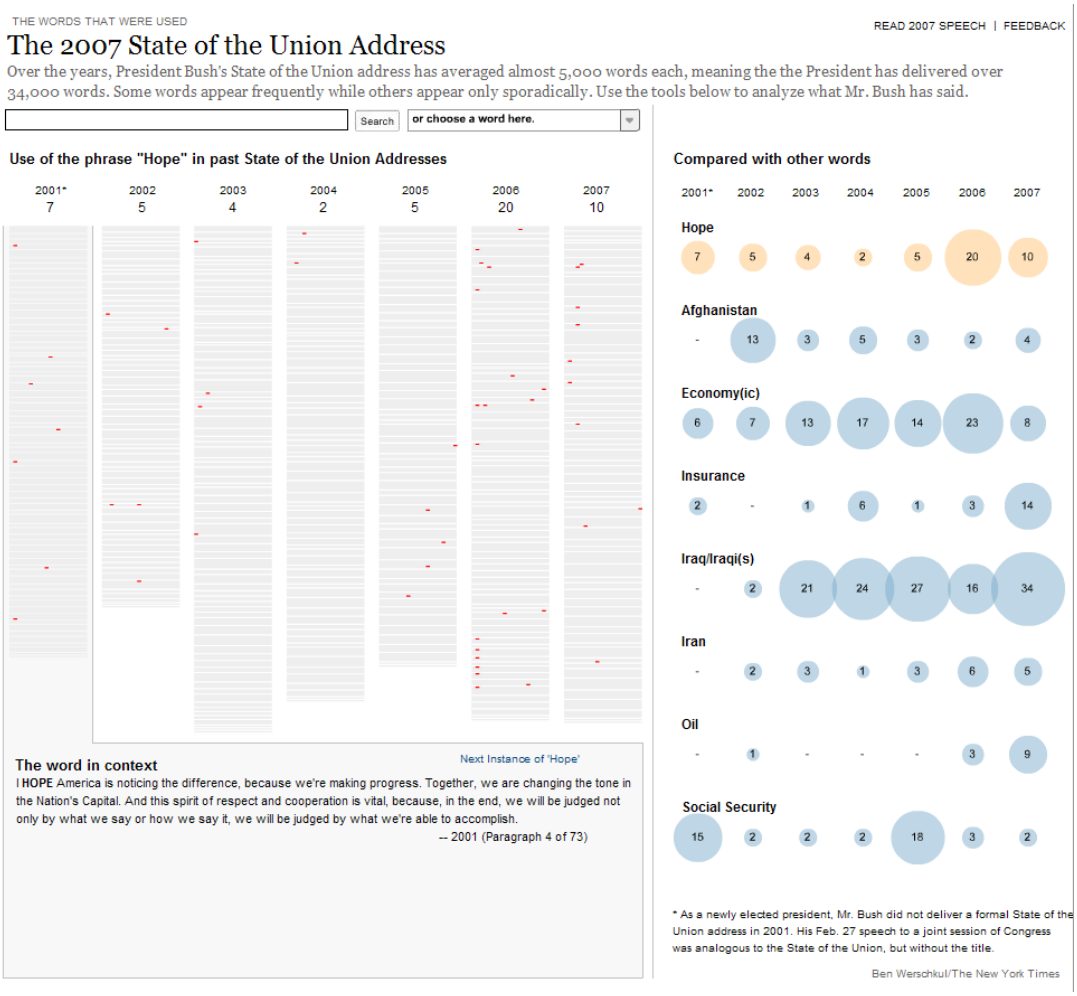
FIGURE 3.7: Linguistic visualizations from the digital humanities community.

repeated sequences and occurrence trends across a document corpus (see Figure 3.7b). Finally, the *WordHoard*, discussed further in Chapter 6 has a simple, minimally interactive word-list interface, but uses more-advanced NLP techniques such as log-likelihood measures to select words of significance to a particular document (Mueller, 2008). Unlike many linguistic visualizations contributed by other communities, all of our examples from digital humanities provide direct, interactive access to the underlying source text for detailed reading.

3.2.5 *Design*

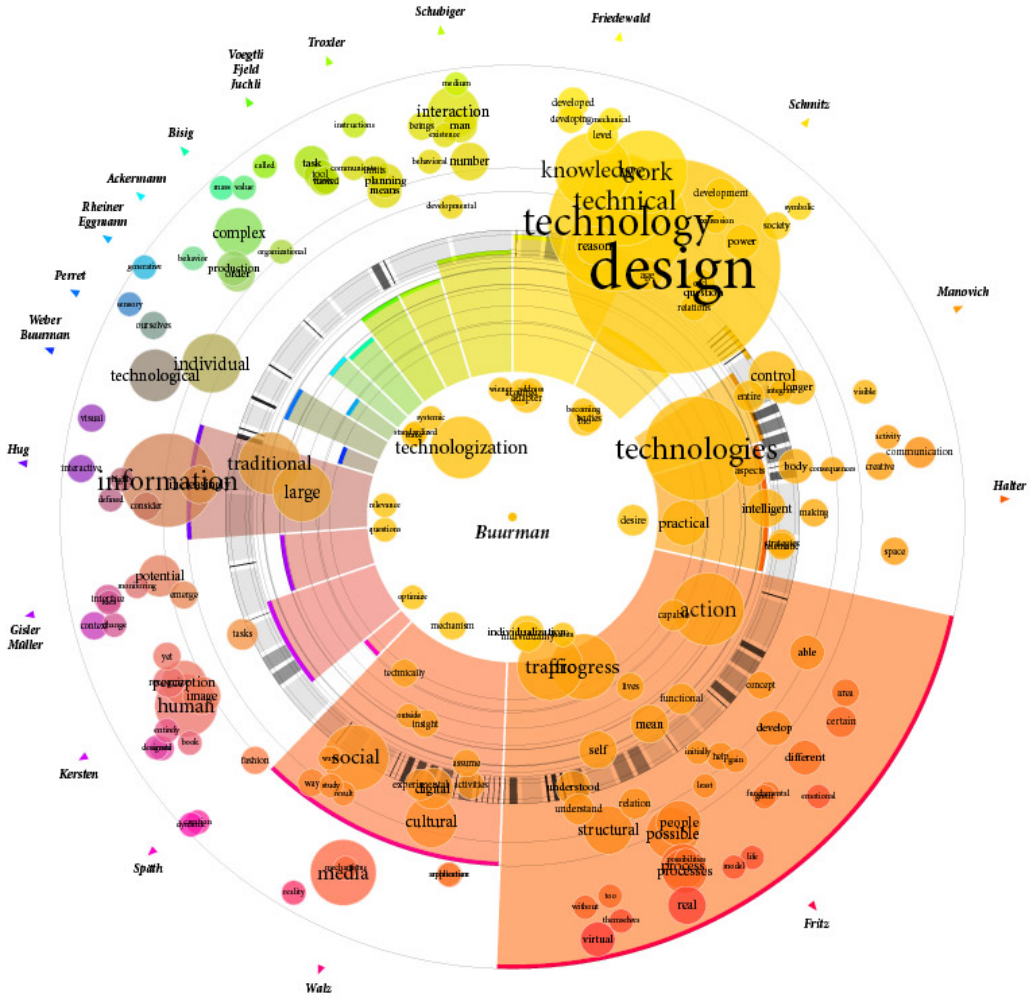
The professional design community produces information visualizations primarily for publication directly on the Web, often for journalistic purposes or commissioned to convey a desired message. While not particularly concerned with generalizable visualization contributions, or formal evaluation, this community strives to create interfaces that are simultaneously beautiful, easy to interpret, and informative (Ericson, 2007). As Norman (2002) claims, "attractive things work better". An example of linguistic visualization from the design community is the *New York Times'* interactive visualization of the text of several years of the *State of the Union Address*. This visualization compresses the transcripts of the addresses into columns by year. When a word is selected, all occurrences are highlighted, using red against grey — a 'preattentive' pop-out. Context is provided through an associated list of popular terms and their relative frequency and a panel to read selected excerpts of the original text (see Figure 3.8a).

The graphical comparative essay visualization commissioned for the book *Total Interaction* (Buurman, 2005) is an example of comparative content analysis from the design community (see Figure 3.8b). The visualization, designed by Rembold and Späth (2006) represents a focus essay as an icon in the center of an annulus, surrounded by the other essays, each represented by coloured arcs of the annulus, with arc length the amount of vocabulary overlap with the focus essay. So, each essay in the printed book is accompanied by an information graphic relating it to the others. The visualization encodes comparative word frequency, text length, paragraph length and structure, and vocabulary overlap between each essay and the central focus text. While the essay visualizations encode a lot of information and are aesthetically compelling, they are not particularly



(a) *New York Times'* interactive visualization of 7 years of State of the Union Addresses (Werschkul and The New York Times, 2007).

FIGURE 3.8: Linguistic visualizations from the design community. *Reprinted with permission.*



(b) Comparative essay visualizations which encode similarities in the lexical content, length, and structure of essays in a book (Rembold and Späth, 2006).

FIGURE 3.8: Linguistic visualizations from the design community (cont'd). *Reprinted with permission.*

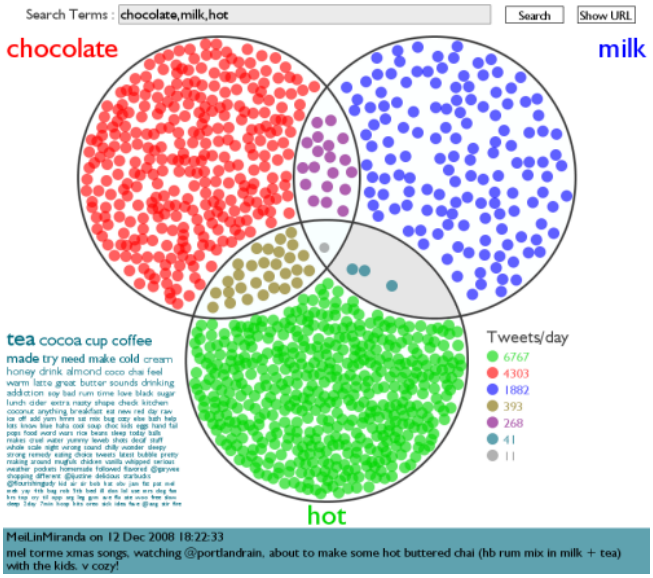
interpretable due to overlapping visual encodings such as hue and value (see §2.3.3).

3.2.6 *Visualization Enthusiasts*

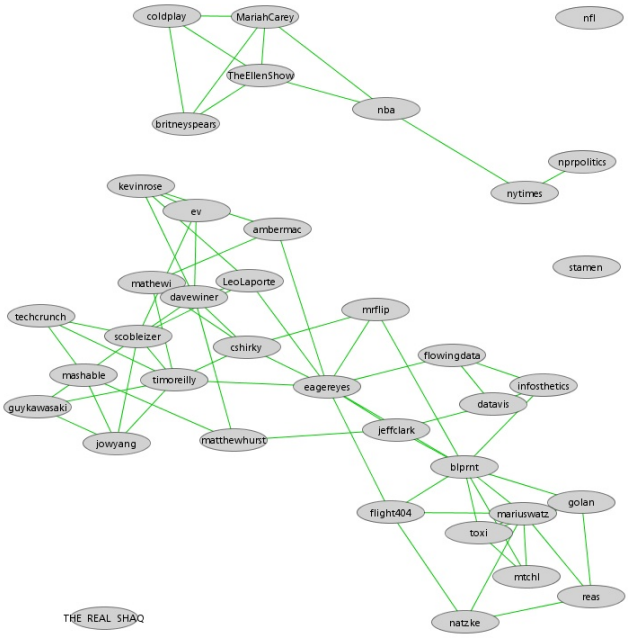
There has been a proliferation of easy-to-learn visualization prototyping tools, designed specifically for Web deployment, including *Processing* (Fry and Reas, 2008), *Adobe Flash*, and open APIs such as the *Google Visualization API* (`http://code.google.com/apis/visualization/`) and the *Raphaël* Javascript library (`http://raphaeljs.com`). In addition, tools for collecting data from the Web are becoming more accessible. Open APIs are enabling anyone with the network access and basic programming skills to collect and visualize data from many linguistic sources. The variety of data types available is continuously growing, and includes social network data and microblog feeds (*Twitter*, `http://twitter.com/twitterapi`), newspaper archives (*New York Times*, `http://developer.nytimes.com/`), thousands of out-of-copyright books (*Project Gutenberg Library*, `http://www.gutenberg.org`), and government data (*e.g.*, *Visible Government*, `http://visiblegovernment.ca` in Canada and the *Sunlight Foundation*, `http://services.sunlightlabs.com/api` in the United States).

It is becoming increasingly easy for information visualization enthusiasts to pick up these tools and create interactive linguistic visualizations to share online. Viégas and Wattenberg (2008) refer to these visualizations by 'nonacademic designers' as *vernacular visualization*: "a visualization which does not come from the visualization community and may violate some of the golden rules of traditional visualization design". Here we are referring to people who create visualizations for fun or personal purposes. Blogger Jeff Clark writes of his motivation for creating linguistic visualizations: "I enjoy discovering the patterns in the apparent chaos of real life data and exploring new techniques for communicating what I discover in a visually compelling manner." (Clark, 2008b, about page). Many of the works created by this community are also examples of *casual infovis*, as they are created for personal enjoyment and based on real-life or personal data (Pousman *et al.*, 2007).

Twitter Venn is an interactive visualization showing Venn diagrams of *Twitter* messages containing up to three specified search terms (Clark, 2009b). By selecting an area of the diagram, a rank-ordered tag cloud of other words in messages from that area is shown. Here the NLP is



(a) *Twitter Venn* of {hot,chocolate,milk} with common collocates of *hot tea* displayed in the lower left (Clark, 2009b).



(b) A social network graph inferred from word frequency similarities amongst various *Twitter* streams (Clark, 2009a).

FIGURE 3.9: Linguistic visualizations from the visualization enthusiasts community. *Reprinted with permission.*

simplistic: word and co-occurrence counting. However, despite simplistic NLP, the intuitive interaction, topical data, and dissemination by the Web make *Twitter Venn* a notable contribution (see Figure 3.9a). Another work, *Twitter Stream Similarity*, uses word frequency similarity measures to determine similarity scores between various members of *Twitter* (Clark, 2009a). The linguistic similarity measure was formulated by Clark, and is essentially the cosine similarity between word frequency vectors, a technique well-known in NLP. The similarly scores are used to cluster and connect nodes into an *inferred* social network graph, an example of using linguistic data as a scaffold for determining social relations (see Figure 3.9b).

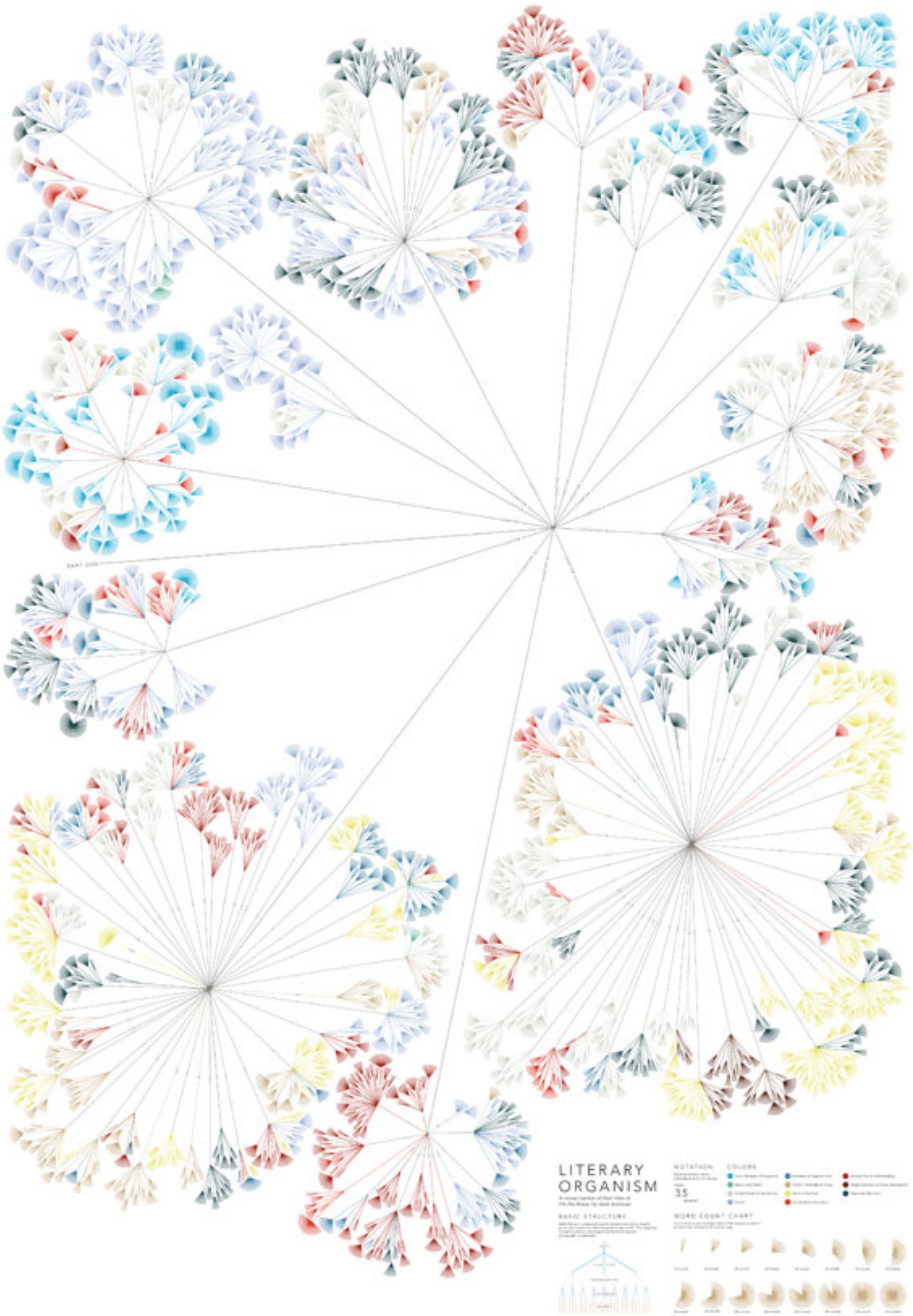
While visualization tools and data streams with high-level and accessible APIs are widely available, the same is not true of services for NLP. Nor do the data provision services themselves offer tools to manipulate the data with NLP algorithms. There are NLP toolkits available for download and use on one's personal computer (*e. g.*, *Natural Language Toolkit* `http://www.nltk.org`); however, these do not fit into the service-oriented Web 2.0 ecology within which the visualization enthusiast community tends to work. Exceptions include the *Alchemy API* (`www.alchemyapi.com`), a paid Web service which provides various NLP algorithms (*e. g.*, named entity and keyword extraction, language identification) and *Open Calais* (`www.opencalais.com`), a free semantic annotation and named-entity extraction service. The dearth of free, general-purpose NLP Web APIs may contribute to the simplistic NLP used in much of the work produced by this community.

3.2.7 *Art*

Defining what constitutes *artistic* visualization is difficult. We will adopt the rather matter-of-fact definition of Viégas and Wattenberg (2007): "Artistic visualizations are visualizations of data done by artists with the intent of making art." Linguistic visualizations created by the art community have been produced both as online, interactive interfaces, and as computer-generated information graphics. *Apartment* (Wattenberg *et al.*, 2004), *The Dumpster* (Levin *et al.*, 2005), and *We Feel Fine* (Harris and Kamvar, 2006) are interactive visualizations of the personal emotions expressed online. *Apartment* gathers data by direct contributions from the viewers while the others use extensive crawling of blogs to match statements about breakups,

(a) *We Feel Fine* is an online interactive artwork which visualizes feelings statements scraped from blogs (Harris and Kamvar, 2006).

FIGURE 3.10: Linguistic visualizations from the artistic community. *Reprinted with permission.*



(b) *Literary Organism* uses a set of rules to divide and categorize the text of a book, then produces an organic-looking tree graph which reveals the structures and themes in the text (Posavec, 2008).

FIGURE 3.10: Linguistic visualizations from the artistic community (cont'd). *Reprinted with permission.*

and feelings, respectively. *We Feel Fine* extracts blog mentions of 'feeling' and associates the text of the statement with extracted demographics about the author and related metadata gathered from other Web sources. The visualization is presented in six *movements* of varying abstraction. One of the movements is based on faceted navigation (Hearst, 2006), and with it one can explore, for example, the recent feelings of women aged 20–25 who live in a place where it is currently raining (see Figure 3.10a).

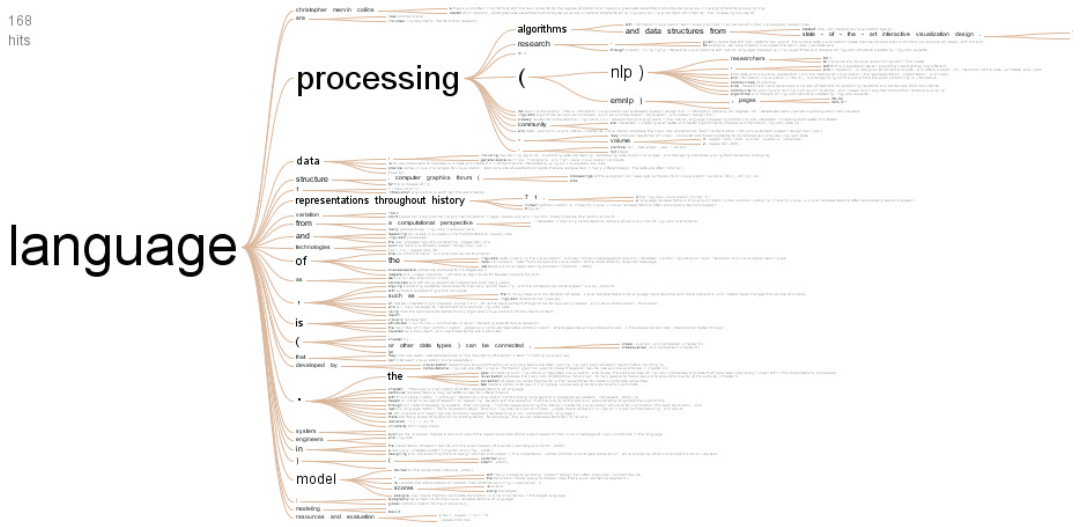
Literary Organism is an example of an artistic visualization of the contents of a book, generated by a text analysis algorithm which breaks the text into chapters, paragraphs, sentences, and words (Posavec, 2008). Sentences are then colour-coded by regular-expression-based theme matching (themes were determined by the artist). The organic-looking tree visualization is an informative form of art from which the length, structure, and thematic patterns in a text can be read (see Figure 3.10b).

3.2.8 *Information Visualization*

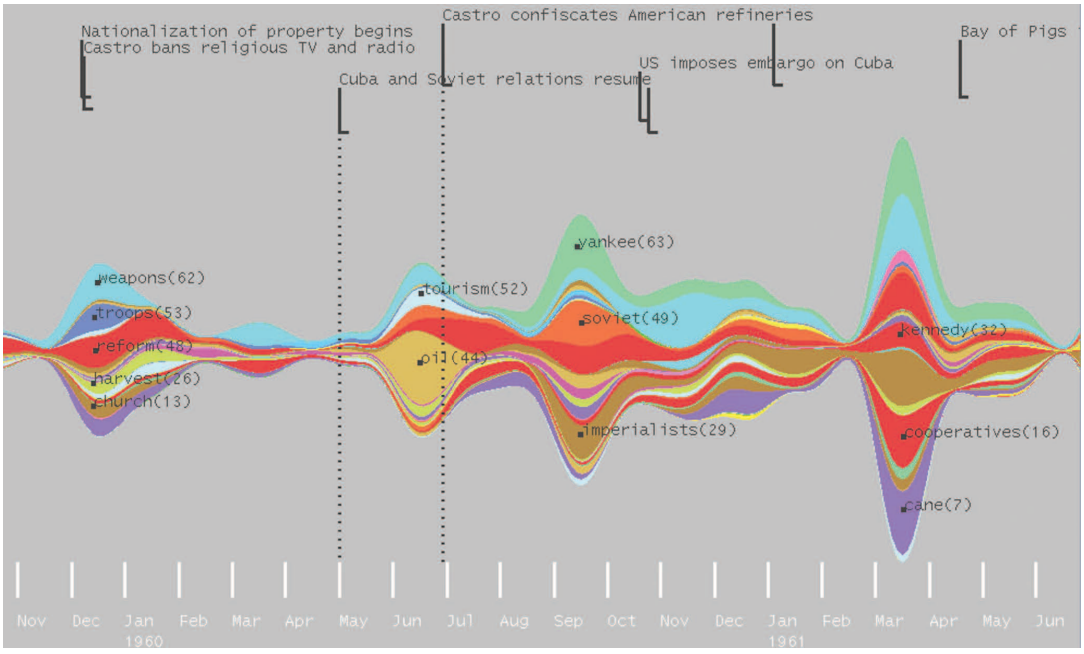
Information visualization researchers are generally concerned with creating interactive visualizations that accomplish at least one of the following:

1. revealing information that was not previously visible or enabling new insights about data,
2. improving the clarity and readability of known representations,
3. improving layout and rendering efficiencies of known techniques,
4. improving usability of or analyst satisfaction with known techniques,
5. presenting new and generalizable techniques for representing, presenting, or interacting with data.

Contributions of each of these types have come from linguistic visualizations published in this community. For example, the *Word Tree* is essentially an interactive form of the KWIC display, improving usability and satisfaction with the concordance technique (contribution type 4) (see Figure 3.11a). On using *Word Tree*, a blogger described the interaction capabilities as enabling exploration of his data from overview to detail, allowing him to discover previously unrecognized inconsistencies (Wattenberg and Viégas, 2008). As a second example, the *Theme River* visualization of streaming news text (Havre *et al.*, 2002) (see Figure 3.11b) contributed the general technique of stream graphs (contribution type 5), which was



(a) A *Word Tree* interactive visual concordance of this dissertation (Wattenberg and Viégas, 2008). *Reprinted with permission.*



(b) *Theme River*, an interactive visualization of streaming news text, contributed the general visualization technique of stream graphs (Havre *et al.*, 2002). © *IEEE. Reprinted with permission.*

FIGURE 3.11: Linguistic visualizations from the information visualization community.

later improved and applied to other data domains (Byron and Wattenberg, 2008).

3.2.9 *Visual Analytics*

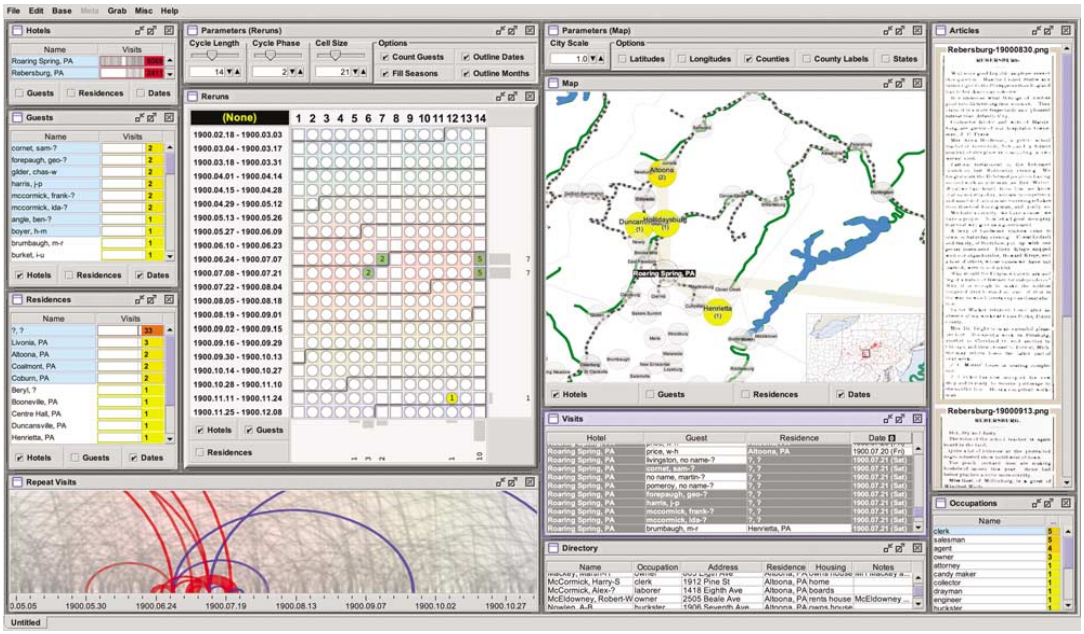
Visual analytics is an outgrowth of information visualization concerned specifically with visual tools to support analytical reasoning processes, such as synthesizing information and deriving insight from massive, ambiguous, conflicting, and multimedia data. One type of data of interest is text, particularly named entities and how they relate to other types of data. Weaver *et al.* (2007) investigate historical hotel registers, analyzing the names contained within, and correlating them with local news articles, geographical maps, and other data sources (see Figure 3.12a). Jigsaw (Stasko *et al.*, 2007) is also based on multiple, coordinated views of disparate data (see Figure 3.12b). In that system, named entities are extracted from documents and the relations between sets of entities are revealed.

3.2.10 *Summary*

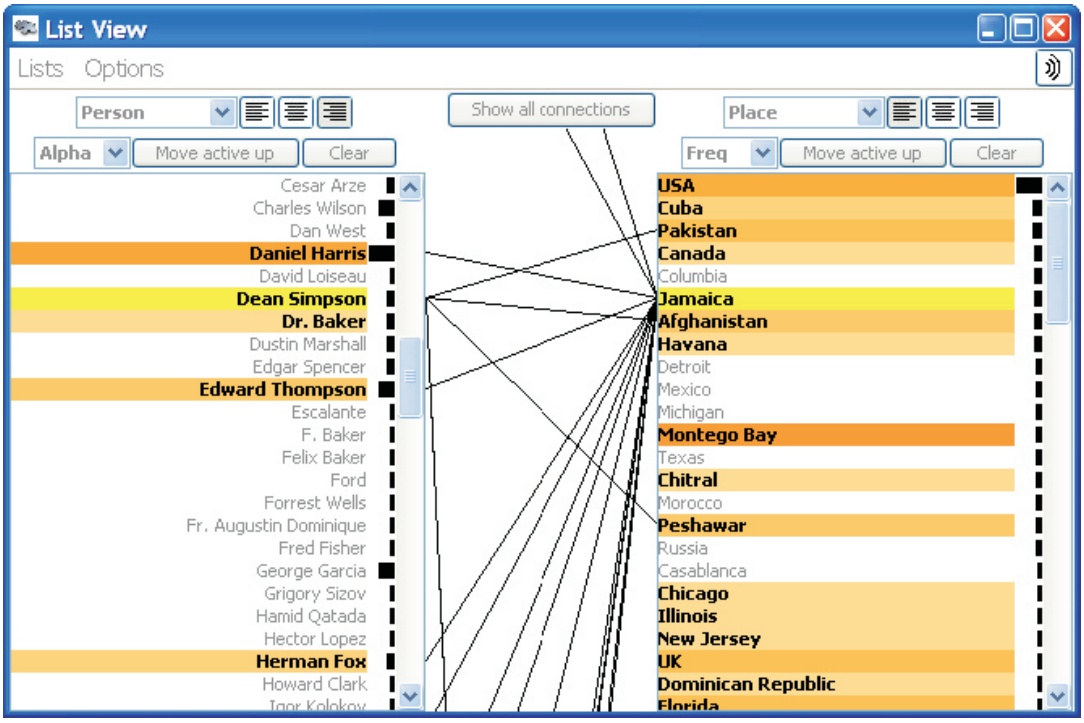
The *space of linguistic visualization* includes contributions from many varied communities of practice. Some communities take a linguistically sophisticated but interactively weak approach (CL and NLP), others are focused on leveraging interaction to enhance analytical power (HCI, InfoVis, Visual Analytics) but do not often connect the interaction to linguistically sophisticated data models. Still others (artists, some designers, enthusiasts) are not necessarily aiming to support analysis or discovery, but rather to use visualization as an expressive medium. Through investigating the variety of communities, we have simultaneously introduced many of the most significant contributions to the field of linguistic visualization.

3.3 TARGET AUDIENCE

We divide the potential audiences for linguistic visualization based on general linguistic expertise and the amount of effort they may be willing to put into learning an interface. We will discuss three groups: the general public, domain experts interested in the *content* of a particular type of



(a) The interlinked views of the hotel visitations visualization (Weaver *et al.*, 2007). © Palgrave Macmillan. *Reprinted with permission.*



(b) Jigsaw list view relating people and place names extracted from a set of documents (Stasko *et al.*, 2007). © *IEEE. Reprinted with permission.*

FIGURE 3.12: Linguistic visualizations from the visual analytics community.

linguistic data, and linguistic, CL, and NLP researchers. Figure 3.13 provides an overview of the space of linguistic visualization, grouped according to target audience.

3.3.1 *General Public*

When linguistic visualizations are designed for the general public, they are often deployed online or as an installation in a gallery, museum, or library. There are examples of both analytically-focused (*e.g.*, Viégas *et al.*, 2007; Weskamp, 2004) and artistic (*e.g.*, Harris and Kamvar, 2006; Levin and Lieberman, 2004) linguistic visualizations online. Similarly, analytical (*e.g.*, Alexander *et al.*, 2007) and artistic (*e.g.*, Legrady, 2005; Wattenberg *et al.*, 2004) linguistic visualizations have been exhibited as installations. Hinrichs *et al.* (2008) recommend that visualization for public spaces such as museums should embody the "walk-up-and-use" principle. The design community similarly acknowledges the general public is unlikely to invest time to read instructions for an online visualization (Ericson, 2007). The initial display should be attractive and interesting, and the interaction should be obvious. The general public will likely be fluent in the language of the linguistic data underlying the visualization, but may not be knowledgeable about or interested in either linguistics or novel interaction and visualization techniques.

Many Eyes is a suite of online text visualizations, targeted at the general public, which have become very popular. Members of *Many Eyes* can upload their own text, visualize it using several different techniques, interact with the visualization, save preferred views for sharing, and annotate their own visualizations or those of others. The available linguistic visualizations include: tag clouds (one word, two word, and comparative) (Viégas *et al.*, 2007), *Wordle* (an aesthetically-rendered tag cloud) (Viégas *et al.*, 2009), *Word Tree* (Wattenberg and Viégas, 2008) (see § 3.2.8 and Figure 3.11a), and *Phrase Nets* (van Ham *et al.*, 2009). Backing these visualizations are simple NLP techniques: tag clouds and *Wordle* uses word-counting and stemming (removal of suffixes), the *Word Tree* builds and displays suffix trees (Stephen, 1994, ch. 4), and *Phrase Nets* use regular expression matching. While *Many Eyes* was envisioned for *casual infovis* and to support *collaboration* amongst the general public, it has also been used in unexpected ways by several linguistic researchers (Danis *et al.*, 2008; Wattenberg and Viégas, 2008). Viégas *et al.* (2009) describe the factors

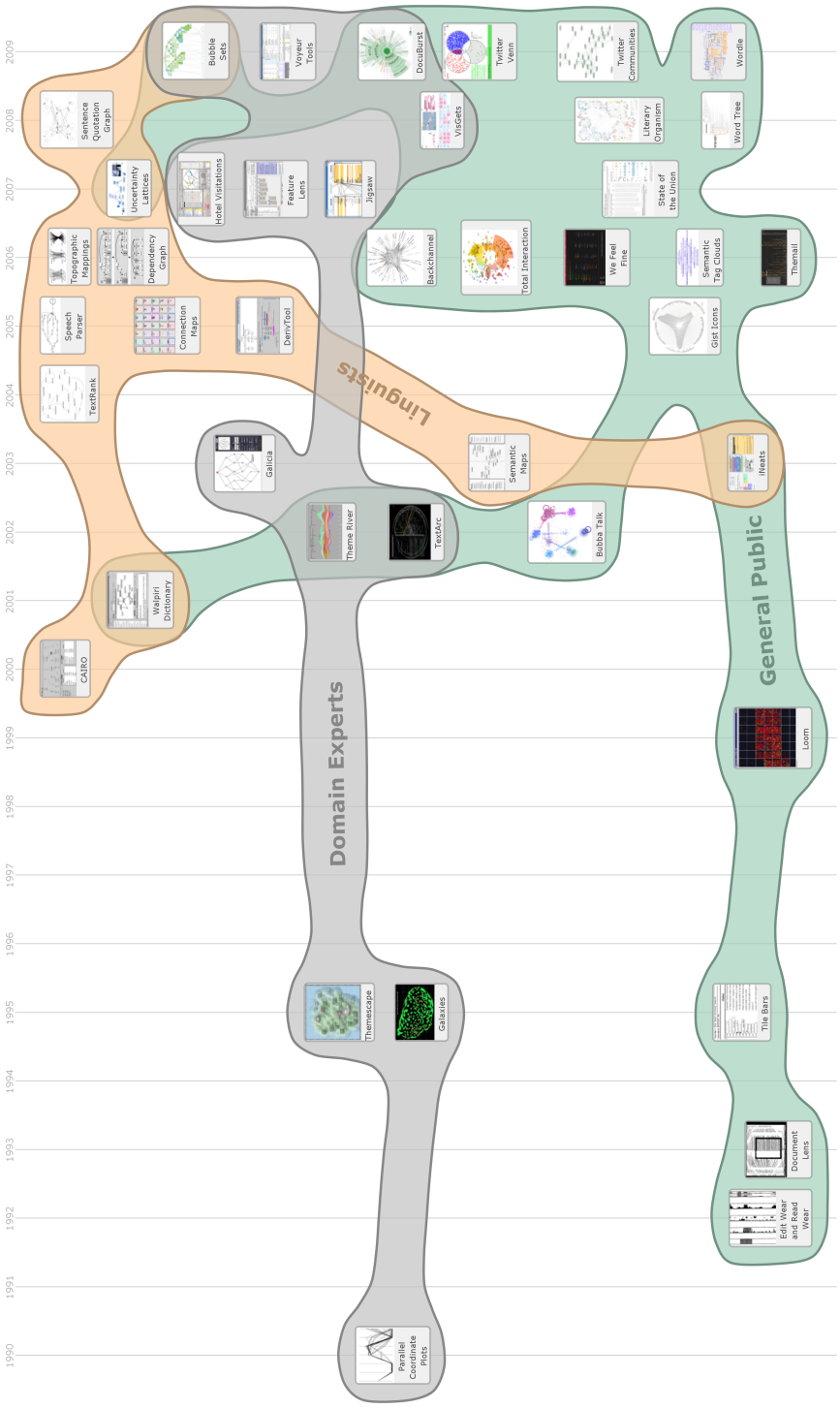


FIGURE 3.13: A Bubble Sets timeline illustrating the related work grouped by target audience.

FIGURE 3.14: The *Wordle* visualization, here depicting the text of this dissertation, was created to target the general public (Feinberg, 2008; Viégas *et al.*, 2009). *Reprinted with permission.*

which made *Wordle* attractive to the general public (over 600,000 'Wordles' were created in under a year). The public craves aesthetics, a feeling of creativity, and engagement with the personal nature of data — despite an acknowledged lack of the traditional supports for analysis, such as interactive elements and clear data encodings, 'Wordle' remains popular. Figure 3.14 shows a *Wordle* of the text of this dissertation.

The previously described *State of the Union* visualization also exemplifies the "walk-up-and-use" principle for general public visualizations (Werschkul and The New York Times, 2007) (see §3.2.5 and Figure 3.8a).

3.3.2 *Domain Experts*

Domain experts are people with expertise and interest in a particular type of linguistic data. Examples of domain experts include political

scientists (transcripts of government proceedings, texts of legislation), legal scholars (transcripts of courtroom discussions, written decisions on cases), historians (news archives, records of personal communication), military intelligence analysts (recorded conversations and correspondence).

The humanities scholars targeted by *Feature Lens* are a group of *domain experts* (Don *et al.*, 2007) (see §3.2.4 and Figure 3.7b). The *Feature Lens* visualization is visually and functionally similar to the *State of the Union* visualization (see §3.2.5 and Figure 3.8a) from the design community. However, where *State of the Union* is based only on word occurrence counts, *Feature Lens* is built upon longer patterns, providing added complexity, but also greater analytical power. The interaction is also different. Where *State of the Union* is intentionally simple, offering only one search box and a drop-down list of suggested words, *Feature Lens* provides long lists of patterns and parameter-tuning options. The humanities scholars targeted by *Feature Lens* may be willing to invest time into learning an interface if the interface complexity rewards them with increased opportunities for insight.

The hotel records visualization from the visual analytics community (see §3.2.9 and Figure 3.12a) is targeted at historians, another form of domain expert.

3.3.3 *Linguistic Researchers*

On the other end of the spectrum sit linguists, computational linguists, and natural language engineers. This group can be considered a specialization within *domain experts*, where the domain is language itself. Members of this community are strongly interested in language, and will likely be eager to interact with and explore linguistic data. They would invest time in learning to use a new system if it had potential to benefit them in the long run. We identify two main goals of this target audience. The first, of interest to linguists and computational linguists, is discovering new insights about language, such as linguistic relationships (see §3.2.1 and Figure 3.3b) or language evolution patterns (*e.g.*, Brighton and Kirby, 2006; Kempken *et al.*, 2007). The second, of interest to NLP researchers, is to analyze and improve NLP algorithms and resources. For example, the previously mentioned *DerivTool* was created to help diagnose problems with an MT system (DeNeefe *et al.*, 2005) (see §3.2.1 and Figure 3.5). Similarly, the *Constellation* visualization was developed to assist the creators of

the *MindNet* semantic network to visualize their data in order to improve their algorithms (Munzner *et al.*, 1999).

3.3.4 *Collaboration*

An important consideration which cuts across the target audience dimension is the number of people interacting with the visualization. Visualizations can be designed for a single analyst, for example the previously mentioned *iNeATS* multi-document summarization (see §3.2.2 and Figure 3.4b). They can also be designed for multiple people working together asynchronously or synchronously, co-located or at a distance.

The dynamics of distant, asynchronous (Heer, 2008) and co-located, synchronous (Isenberg *et al.*, 2008) collaboration over information visualization have been recently studied. The annotation capabilities of the visualizations within *Many Eyes* are examples of asynchronous collaboration at-a-distance. There are many synchronous at-a-distance collaborative visualizations which rely on linguistic input, but use it only as a signal of social connections (*e. g.*, Erickson *et al.*, 2006; Erickson and Laff, 2001; Tat and Carpendale, 2002; Viégas and Donath, 1999). These *social visualizations* do not actually visualize language data (see §3.4.1 for additional discussion). *Cambiera* is a visualization for synchronous, co-located collaborative search which uses language in the presentation of search results and the specification of queries (Isenberg and Fisher, 2009). However, we know of no synchronous visualizations for collaborative analysis of linguistic data.

3.4 PROBLEM AREA

Linguistic visualization has been applied to a wide range of problem areas. Broadly, we categorize these areas as: communication, information retrieval, content analysis, and linguistics and computational linguistics research. There are subcategories within each of these, and the breakdown is not mutually exclusive. Figure 3.15 illustrates the works included in our study of the space of linguistic visualization, grouped by problem area.

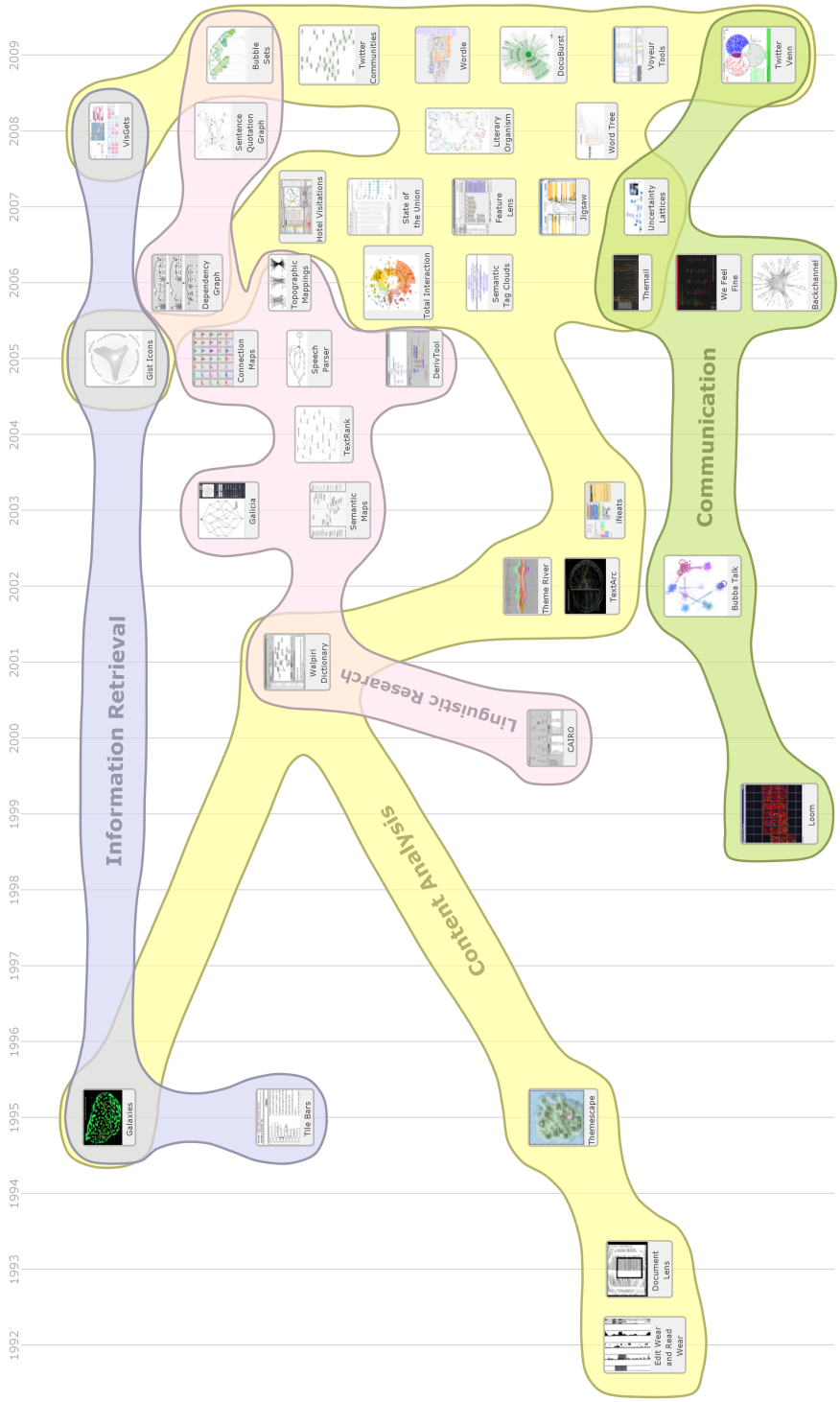


FIGURE 3.15: A Bubble Sets timeline illustrating the related work grouped by problem area.

3.4.1 *Communication*

Language is the main tool of human communication, especially computer-mediated communication, where gestures and expressions are, in the abscence of video, best approximated through *emoticons*. Interests in visualizing communication can be subdivided into real-time communication (conversations) and asynchronous communication (leaving messages for later reading). We will examine examples from each dynamic.

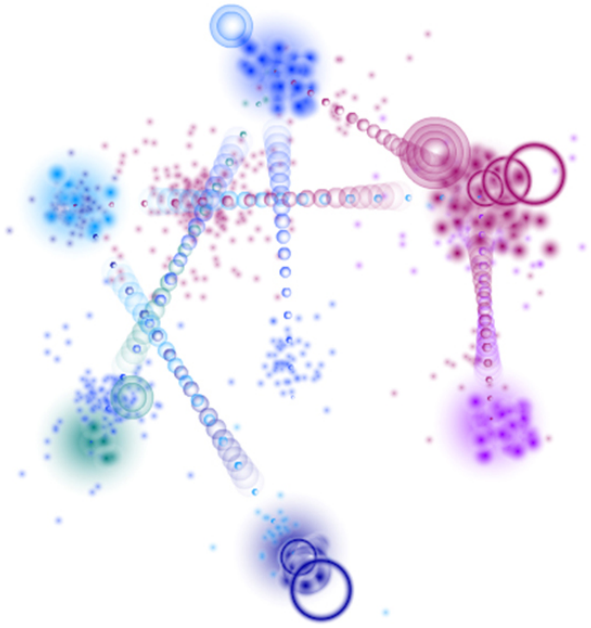
Real-time Communication

Text conversations can take place in near real-time, through technologies such as instant messaging and mobile SMS. The vast majority of work in this area has been in the interest of "providing social context for interaction by providing cues about users' presence and activities" (Erickson *et al.*, 2006). These systems reveal real-time and historical online interactions between participants, while ignoring the content of the messages (*e.g.*, Donath and Viégas, 2002; Erickson *et al.*, 2006; Erickson and Laff, 2001). Therefore we do not classify them as linguistic visualization.

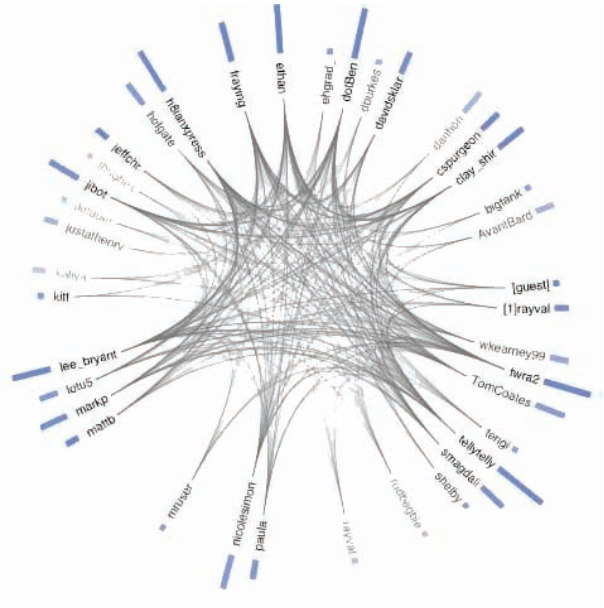
Tat and Carpendale (2002) focus more on the content of online chat in their prototype *BubbaTalk*, illustrating volume of text, timing of messaging, and the use of messaging conventions such as all-uppercase for *shouting*. The visualization itself does not display words, but rather uses abstract representations to convey the dynamics and general character of the conversation as based on several surface features of text (see Figure 3.16a). Similarly, *Backchannel* (Stamen Design, 2006) provides a real-time view of the volume and direction of message passing in an IRC channel (see Figure 3.16b). Both *BubbaTalk* and *Backchannel* provide up-to-date views in the context of the recent history of the conversation, but are not linked with any NLP. Visualizations of the history of instant messaging chat have also been reported (Tat and Carpendale, 2006), but as they present a cumulative view of the history, they are more closely associated with asynchronous communication visualizations.

Asynchronous Communication

Asynchronous communication is communication in which there is a time delay between a message being sent and received. This includes the text-based services that store persistent messages, such as email, newsgroups,



(a) *BubbaTalk* represents six conversants by large coloured clouds arranged around a circle. Messages flow between conversants, leaving traces of their paths. Words written by each participant hover around their cloud, and simple features of the text are encoded using symbols (Tat and Carpendale, 2002).



(b) Backchannel (Stamen Design, 2006). Messages flowing between participants in an IRC room are visualized as increasingly darkly shaded edges. Total volume of text per participant is shown as a blue bar beside the name.

FIGURE 3.16: Visualizations of real-time communication. *Reprinted with permission.*

Web forums, blogs, and co-edited wikis. The line is blurring between synchronous and asynchronous: services such as *Twitter* provide rapid dissemination of short messages which are often replied to immediately (synchronous), but are also archived in a history for future reading and reflection (asynchronous). As the majority of the published visualizations of *Twitter* data aggregate and analyze the historical data, we will discuss this form of data under content analysis (§3.4.2).

Large-scale, many-to-many discussions, such as *Usenet* and *Google Groups*, present opportunities for diverse conversation but also pose challenges to finding valuable information among the thousands of messages posted daily. Work in this area is generally focused on the activity and connections between participants, not the linguistic content of the postings. For example, Viégas and Smith (2004) and Smith and Fiore (2001) describe navigation tools designed around authorship data. These systems aim to graphically represent the population of authors in a particular newsgroup, making it easy to discover the most active authors by number of messages and period of time. Viégas *et al.* (2004) uses the history of changes within *Wikipedia* entries to track patterns of collaboration and conflict over co-authored documents.

An example of a visualization which is more focused on the linguistic content of the messages is the Loom (Donath *et al.*, 1999), which scores newsgroup messages using a set of regular expression rules to roughly match emotion (*e.g.*, profanity=anger) and then visualizes emotional content in a matrix plot (see Figure 3.17).

3.4.2 *Content Analysis*

We choose the generic name *content analysis* intentionally to be inclusive of approaches to visually analyzing many types of text repositories: blogs, newspapers, literature, patents, *etc.*, with the goal of understanding the content. Types of data one may seek to visualize for content analysis include emotional content (Gregory *et al.*, 2006), themes (Wise *et al.*, 1995), relations amongst named entities (Stasko *et al.*, 2007; Weaver *et al.*, 2007) (see Figure 3.12), lexical choice, writing style, and authorship (*e.g.*, Kjell *et al.*, 1994; Plaisant *et al.*, 2006), and patterns of repetition (Eick, 1994; Wattenberg, 2002; Wattenberg and Viégas, 2008) (see Figure 3.11a).

In the fields of CL and NLP, it is common to refer to a large collection of text documents as a *corpus*, which is used to train a model (*e.g.*, Collins,

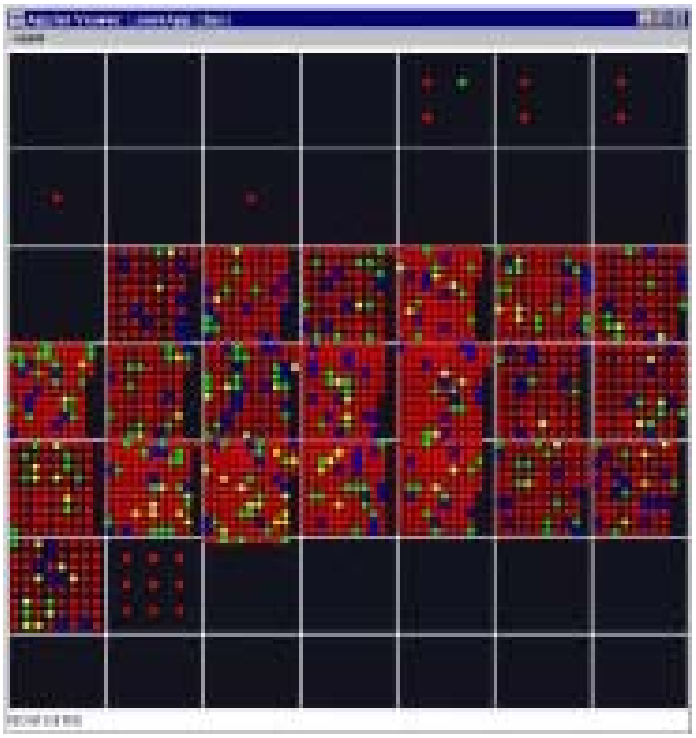


FIGURE 3.17: Loom visualization of emotion in asynchronous communication (Donath *et al.*, 1999). © *IEEE. Reprinted with permission.*

2004). However, we mean corpus in the general sense of "collection of documents" here, not necessarily as linguistic resource.

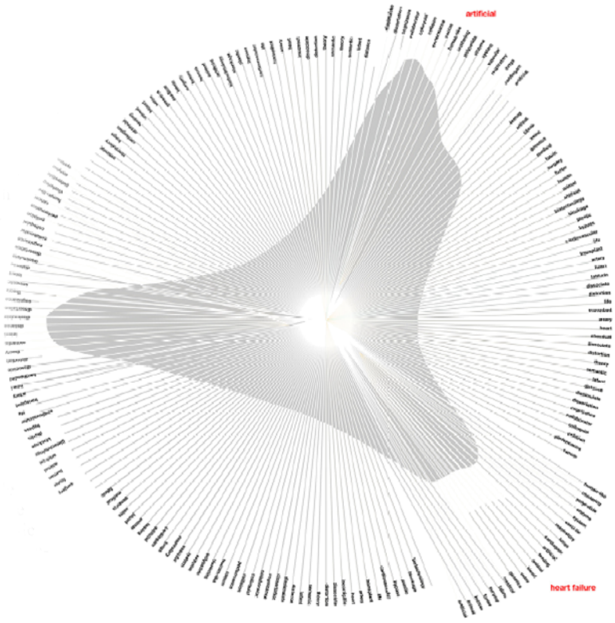
Linguistic visualization provides for content analysis in three main ways: by visualizing data about individual documents, by visualizing relationships between documents in a corpus (corpus overview — discrete), and by visualizing aggregate data about a corpus or a subset (corpus overview — continuous). These subcategories are related in that visualizations of individual documents may be used as small multiples (§2.3) to provide a discrete corpus overview, or a semantic zoom operation may adjust the granularity of the view, merging documents into a continuous corpus overview.

Individual Documents

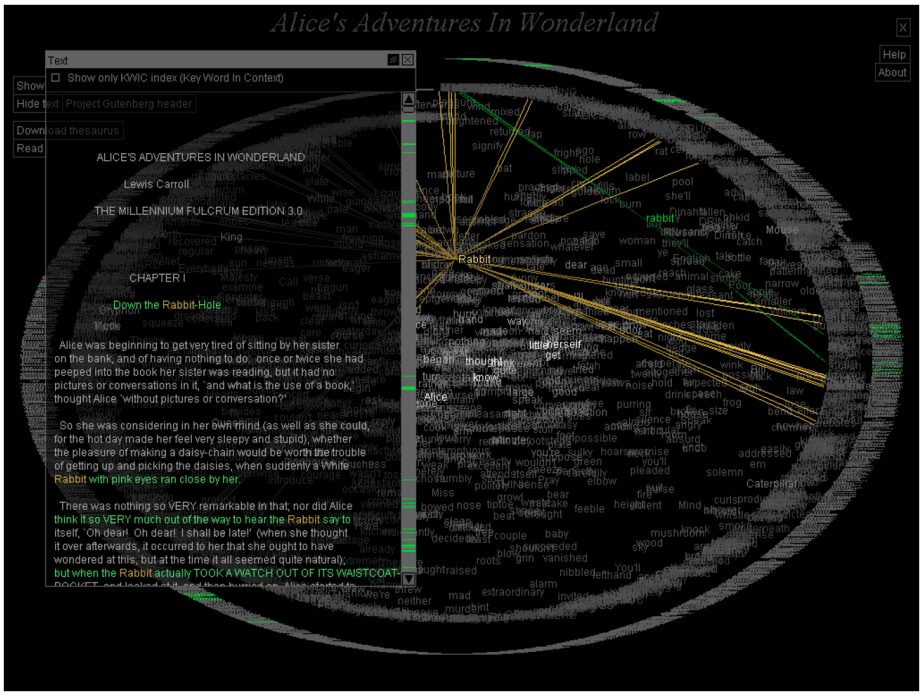
Visualizations of individual document content take two common forms: synoptic visualizations for quick overviews and visualizations specialized for discovering patterns within and between documents. In the remainder of this section we will review significant examples in each of these categories, describing how their feature sets compare.

Synoptic visualizations of text most often use a selected subset of the language to create a glyph based on word occurrence counts. Glyphs are then combined in small multiples visualizations to perform comparisons. Glyph techniques include *Starstruck* (Hetzler *et al.*, 1998b), which creates glyphs by arranging lines of varying length in a circular pattern, and *Gist Icons* (DeCamp *et al.*, 2005), which builds on this idea by drawing a smoothed contour around this pattern (see Figure 3.18a). The vocabularies used are either restricted small-sized user-selected term sets (*Starstruck*) or larger, automatically selected sets using latent semantic indexing (*Gist Icons*). The selection of word sets with latent semantic indexing quickly finds a reasonably sized set of terms related to concepts of interest; however, to the viewer of the visualization, the selection and grouping of terms can sometimes appear non-intuitive. The high number of dimensions in the *Gist Icons* may exceed the viewer's ability to process precise meaning from shape, which is characterized as quantitative "with effort" in Table 2.1.

Other synoptic visualizations of document content use space-filling techniques to provide an overview of the vocabulary a single document, such as tag clouds and *TextArc* (Paley, 2002). *TextArc* is a visualization which lays out sentences clockwise around a circle, and words are placed



(a) Glyph-based document content visualization with *Gist Icons* (DeCamp *et al.*, 2005).



(b) A *TextArc* visualization of *Alice in Wonderland*. Instances of 'rabbit' are highlighted in the overview and detail (Paley, 2002).

FIGURE 3.18: Visualizations of document content. © *IEEE. Reprinted with permission.*

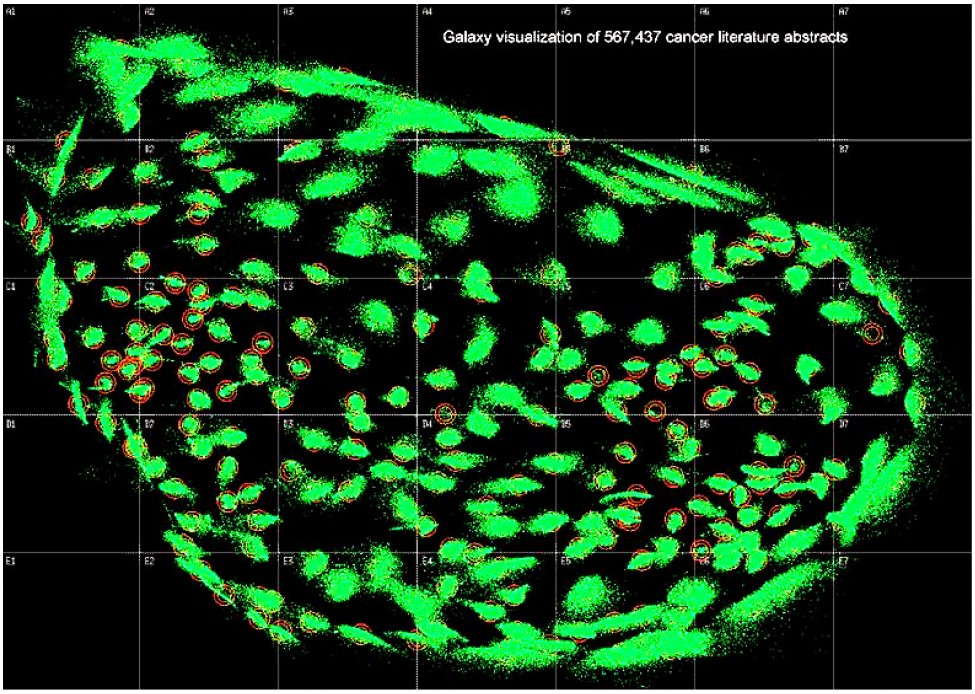
within the circle, closest to their places of appearance (see Figure 3.18b). Thus a common theme, found throughout a text, will be placed near the middle of the circle while a word occurring mostly near the middle of the text will be close to the 6 o'clock position. A linked view provides the ability to read the source document in context.

A second category of document visualizations is those created to reveal patterns within texts. The previously discussed *FeatureLens* (Don *et al.*, 2007) (see Figure 3.7b) suggests repeated phrases which may be of interest, and visualizes analyst selections, while *Arc Diagrams* (Wattenberg, 2002) provide an overview of repetition throughout a document. Other pattern-based visualizations focus on distributions of significant features in a document such as emotion (*e.g.*, extensions (Oelke *et al.*, 2008) of *TileBars* (Hearst, 1995)), or hand-annotated properties (*e.g.*, *Compus* (Fekete and Dufournaud, 2000)). The *Word Tree* (Wattenberg and Viégas, 2008) (see Figure 3.11a) is an example of a pattern-based visualization focused on repetition in context. While *Word Trees* provide a unique view on repetition, overall word frequencies and full document overviews are not visible.

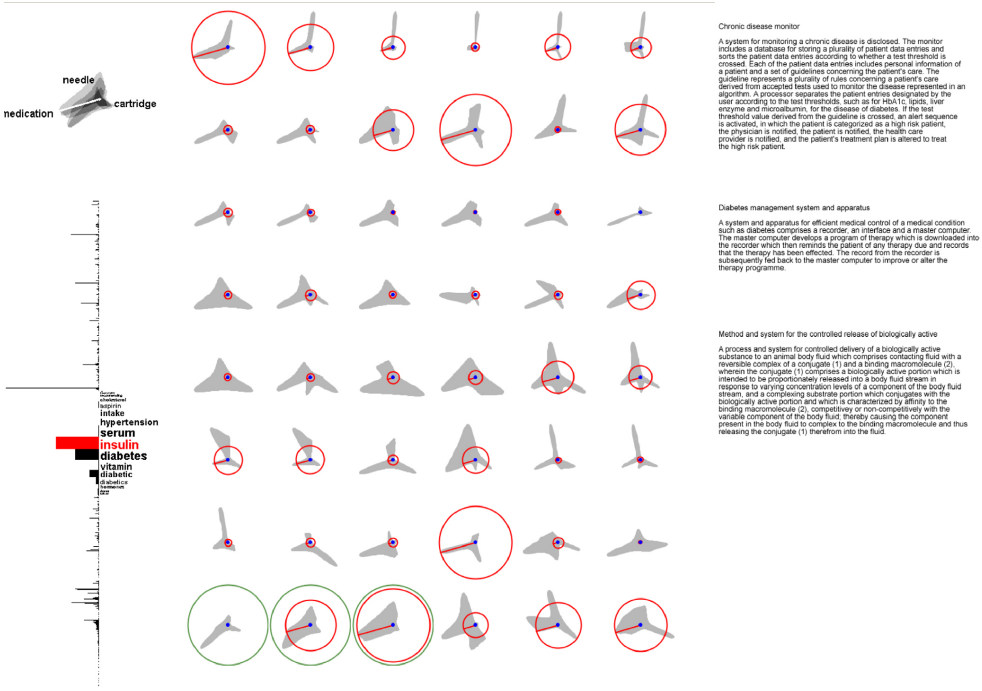
Discrete Corpus Overview

Document corpora can be visualized by a systematic and meaningful display of their members. Characteristics of documents in the corpus, such as their relationship to other documents, their length, or keywords, have been encoded visually. An example of a discrete corpus overview is *Galaxies*, based on the familiar visual scene of a starry night sky (Wise *et al.*, 1995). Galaxies are built up as an emergent form of blue-green points (*docustars*), each representing a document in the corpus, clustered around orange topic centroids (see Figure 3.19a).

As an extension to the *Gist Icons* idea, DeCamp *et al.* (2005) propose a small-multiples matrix views of icons, one per document (see Figure 3.19b). As comparing the radius at an arbitrary position across multiple contoured glyphs would be difficult, interactive supports provide the additional cognitive aid of brushing interaction — if a word position is selected in one glyph, it is highlighted in all for easier comparison. Also, glyphs may be overlaid to compare their shape.

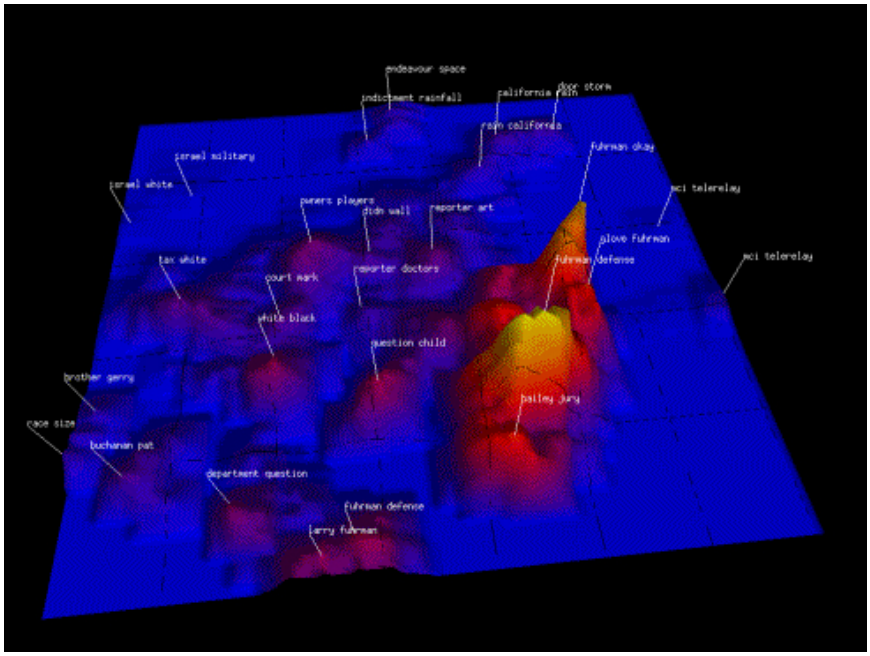


(a) A *Galaxies* view of documents and document clusters in a news text database (Wise *et al.*, 1995).

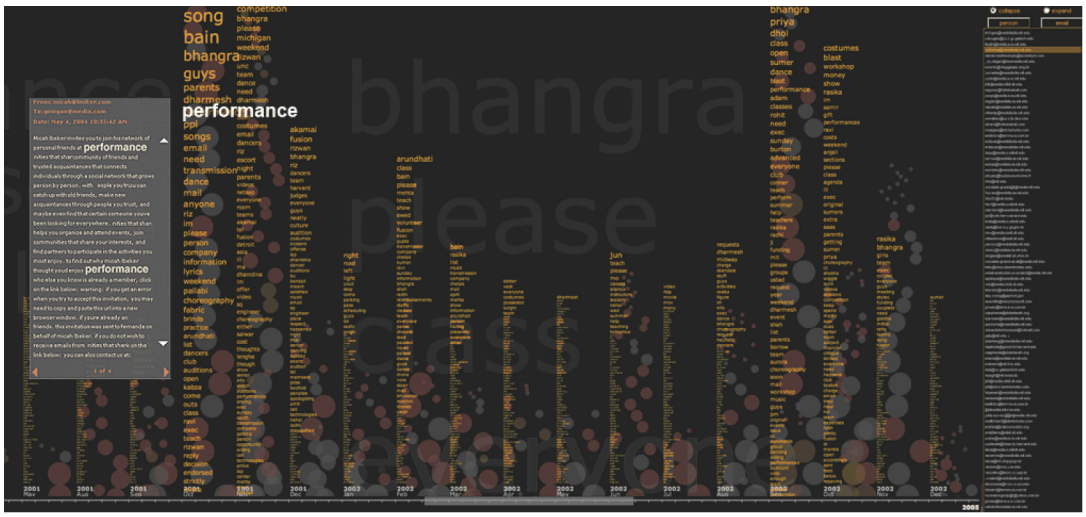


(b) A mock-up of the proposed *Gist Icons* small multiples view of a document corpus (DeCamp *et al.*, 2005).

FIGURE 3.19: Examples of discrete corpus overview techniques. © *IEEE. Reprinted with permission.*



(a) A *Themescape* view of continuous thematic transitions over a spatialized document corpus (Wise *et al.*, 1995). © *IEEE. Reprinted with permission.*



(b) The *Themail* visualization of an email repository (Viégas *et al.*, 2006). © *ACM, Inc. Reprinted with permission.*

FIGURE 3.20: Continuous corpus overview visualizations.

Continuous Corpus Overview

Continuous corpus overview visualizations provide for generalized views of the contents, or subset of the contents, of a corpus. In this approach properties of documents are aggregated to create a high-level summative visualization, whereas in discrete corpus overview each document is represented by an individual mark or glyph. For example, the *Themescape* visualization is created by extracting topic words from individual documents in a *Galaxies* visualization, then smoothing the space to create an undulating 3D surface which represents themes and thematic changes across the spatialized document set (Wise *et al.*, 1995) (see Figure 3.20a). Interactions with a continuous corpus visualization may include *semantic zoom* operations. An example of semantic zoom would be adjusting the granularity of the displayed thematic terms in *Themescape*. Other examples of continuous corpus overviews include the "haystack" aggregation views of email collections in *Themail*. The haystack view reveals frequent words extracted from collections of emails organized by month and year (Viégas *et al.*, 2006) (see Figure 3.20b; also discussed in Chapter 6). The previously examined *Theme River* (Havre *et al.*, 2002) (see Figure 3.11b) is also a form of continuous corpus overview.

3.4.3 *Information Retrieval*

While *information retrieval* may be considered a field separate from both NLP and InfoVis, we include it here as a problem area of interest to linguistic visualization researchers when the corpus to be searched is linguistic in nature. Presentation of search results is closely related to providing corpus overview visualizations for content analysis. Indeed, the discrete corpus overview mode of *Gist Icons* is also a form of linguistic visualization for information retrieval, designed to assist people in finding the document that best matches their interest. *Gist Icons* is conceptually related to *Tile Bars*, an early and well-known example of a linguistic visualization of search results. *Tile Bars* uses rows of horizontal bars, one per search result, to display patterns of relative strength of query terms within search results (see Figure 3.21a). The visual variable of value, which is ordered, is used to encode the strength of the query term in a universally understood gray-scale (Bertin, 1983). The visual variable of position is used to indicate where in a document query terms co-occur. Despite theoretically strong design choices, usability studies show that people

are on average slower when using the interface than with a text-based search (Hearst, 2002). Other deployed visual search interfaces, *e.g.*, *Kartoo* (`http://www.kartoo.com`) have yet to become widely popular. Hearst (2009, ch. 10) offers a comprehensive review of this area and hypothesizes that the lack of uptake of linguistic visualizations of search results may be due to the linguistic visualizations attempting to play double duty as a text to be read and a space to be visually perceived for search.

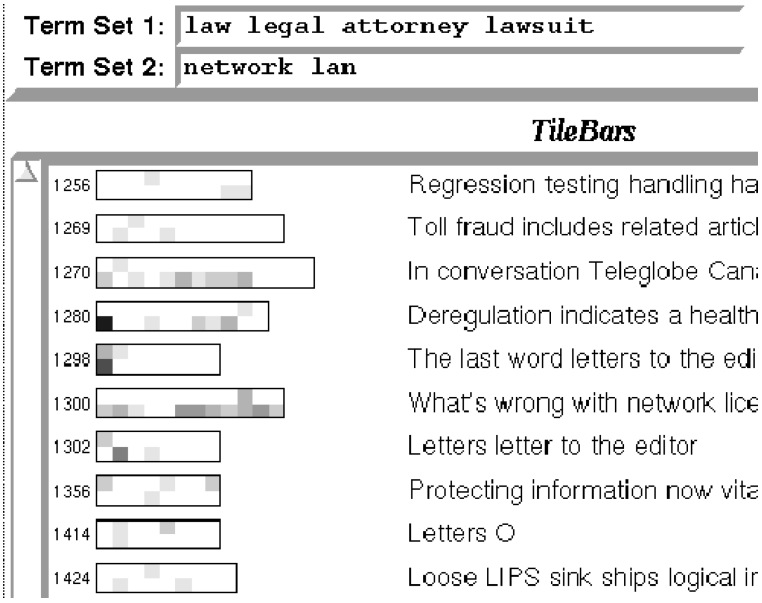
Linguistic visualizations for information retrieval have also been designed for interactively specifying search queries (*e.g.*, Jones and McInnes, 1998). Our *VisGets* interface is an example of a hybrid interface which is both a visualization of the contents of a corpus, and a faceted visual search interface for query specification. The individual *VisGets* (map, tag cloud, timeline, *etc.*) operate both as linked widgets for query specification and as visualizations of the contents of currently available documents (see Figure 3.21b).

We do not specifically address this problem area in this dissertation, choosing instead to focus on the other three areas and incorporate information retrieval within our projects as needed. Design studies such as DocuBurst (Chapter 5) and Parallel Tag Clouds (Chapter 6) provide information retrieval capabilities through coordinated views used to retrieve a particular document or paragraph of a document.

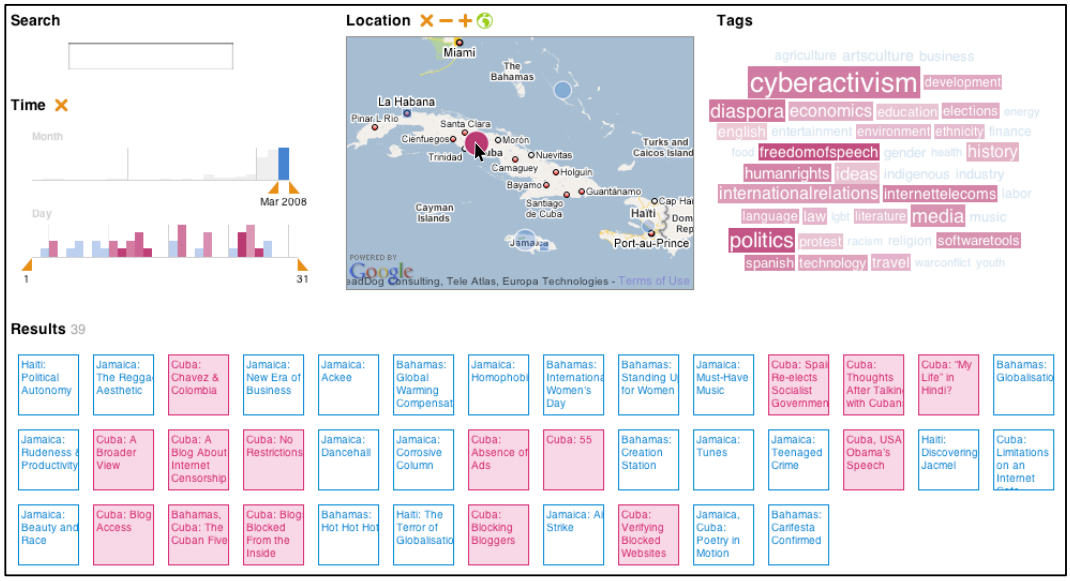
3.4.4 *Linguistics and Computational Linguistics Research*

While the majority of linguistic visualizations published *by* the CL and NLP communities are static *information graphics* designed for communicating theories and results, there are examples of interactive visualizations bridging the *linguistic visualization divide*, providing useful cognitive aids to *support* linguistics, computational linguistics, and natural language processing research. For example, we have already discussed the *DerivTool* (DeNeefe *et al.*, 2005), *CAIRO* (Smith and Jahr, 2000) and *Constellation* (Munzner, 2000) systems (see §3.2.1 and Figure 3.5).

Thiessen (2004) describes another linguistic visualization called connection maps, which was applied to the task of viewing semantic similarity relationships between verbs in English. Verb similarities were calculated using the general feature space of Joanis (2002). This visualization is based on arranging elements of a highly connected graph into a grid format. Each grid element is then subdivided into individual connection maps,



(a) The *Tile Bars* visualization reveals the occurrence and co-occurrence strengths of search terms within documents matching a query (Hearst, 1995). © *ACM, Inc. Reprinted with permission.*



(b) The individual widgets in the *VisGets* interface operate both as corpus overview visualizations and visual query specification widgets (Dörk *et al.*, 2008). *Reprinted with permission.*

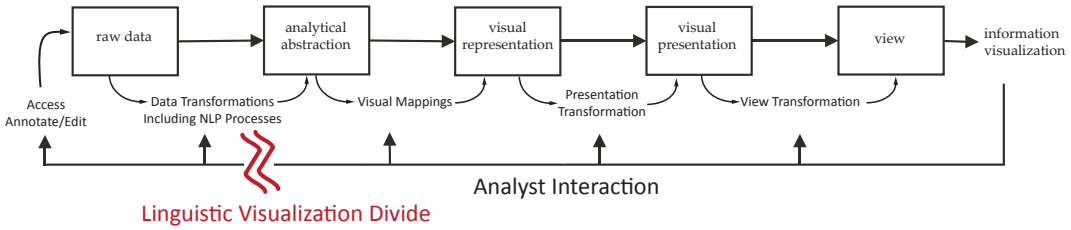
FIGURE 3.21: Visualizations for information retrieval.

FIGURE 3.22: A connection map view of verb classes in English (Thiessen, 2004). *Reprinted with permission.*

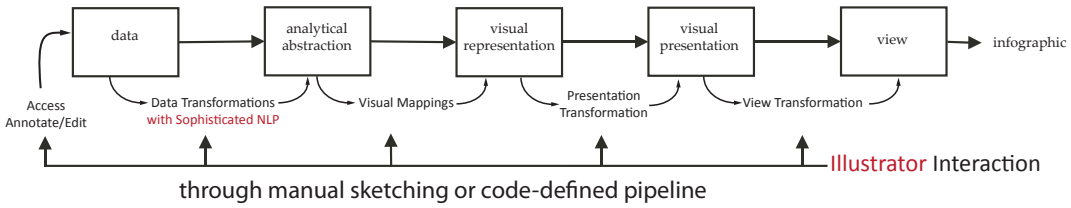
which are a mini-grid representation of the entire grid. A connection between element *A* and element *B* is indicated by placing *B*'s colour in the cell of *A*'s connection map that corresponds to the position of *B* in the main grid. Layout in the connection map is configured to optimally minimize the distance between connected elements. Adjacent connected verbs lead to adjacent coloured squares in the connection maps. By the Gestalt law of proximity, adjacency of coloured squares will be intuitively interpreted as a group, making easy the task of detecting verb clusters. Connection maps are filled for all verbs in the collection, presenting an overall view of the verb set which may enhance the ability of linguistic researchers to discover patterns of relations (see Figure 3.22).

3.5 LEVEL OF INTERACTIVITY

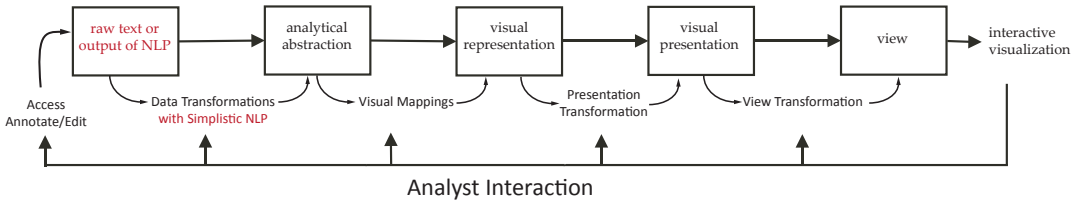
We have proposed that a gulf (the *linguistic visualization divide*) seems to exist in *space of linguistic visualization*: visualization designers do not often connect interaction with *live* NLP, but rather pre-processed data; NLP and CL researchers do not make use of the power of *interaction* when creating visualizations of linguistic data. The result is that viewers of linguistic



(a) The linguistic information divide is manifested in the visualization pipeline as an inability for visualization applications to modify data or apply data transformations using NLP algorithms.



(b) Linguistically sophisticated data transformations and representations are often static infographics.



(c) Linguistically simple data transformations (word-counting) and often highly interactive visualizations.

FIGURE 3.23: The linguistic visualization divide.

visualizations are often not able to access or change underlying data or data transformations (see Figure 3.23).

The level of interactivity dimension is related to the *information visualization pipeline* (§2.3.2). For a given visualization, to determine its levels of interactivity, we enumerate the points of the pipeline at which a person can make adjustments: access/annotate/edit data, modify data transformations, modify visual mappings, modify the presentation transformation, and modify the view transformation. The depth-of-interaction is the leftmost (closest to the data) access point available to the visualization user. As a given visualization may support several levels of interactivity, and the details of interaction are not always apparent from research reports, a Bubble Sets diagram of the space of linguistic visualization along this dimension would not be practical. Instead, we will focus on some examples to illustrate how this dimension can be used to describe a visualization.

For example, the previously mentioned linguistic visualization *State of the Union* (see Figure 3.8a) allows for selection of a focus column and focus term, and provides brushing interaction to view the context of word occurrences. These are all changes to the presentation transformation. *State of the Union* does not support data access/annotate/edit, view changes, or modifications to the representational transformation. So, it operates only at the *presentation* level of interactivity.

As a second example of how this dimension applies to the collection of related work, consider the *Word Tree* linguistic visualization of the the *Many Eyes* suite of tools (see Figure 3.11a) (Wattenberg and Viégas, 2008). The basic concept of *Many Eyes* is to provide facilities for the general public to upload data and discuss visualizations. Therefore, *Word Tree* supports data annotation. It does not, however, support data editing. Linguistic datasets in *Many Eyes* are not editable through the visualization or website (they must be re-created to change). *Word Tree* also does not provide an access facility to interactively transition between using the visualization and viewing the underlying source text related to the current view. *Word Tree* provides for modifications to the data transformations through adjusting filters on the data (root tree at a different word). Presentation transformations (highlight items, layout tree left-to-right or right-to-left) are supported, but view transformations are not (the zoom factor is automatically set to fit the tree). Therefore, the levels of interactivity for *Word Tree* are data annotation, data filter, and presentation transformation. The deepest level of interaction for *Word Tree* is at the data annotation level.

3.6 TYPE OF LINGUISTIC RESOURCES

The type of linguistic resources — data and algorithms — backing a linguistic visualization vary from simple word counting to sophisticated statistical models to expert-created linguistic ontologies. Figure 3.24 illustrates the related work in the space of linguistic visualization, grouped by type of linguistic resources used in the visualization. We identify five types of linguistic resource, roughly increasing in linguistic sophistication:

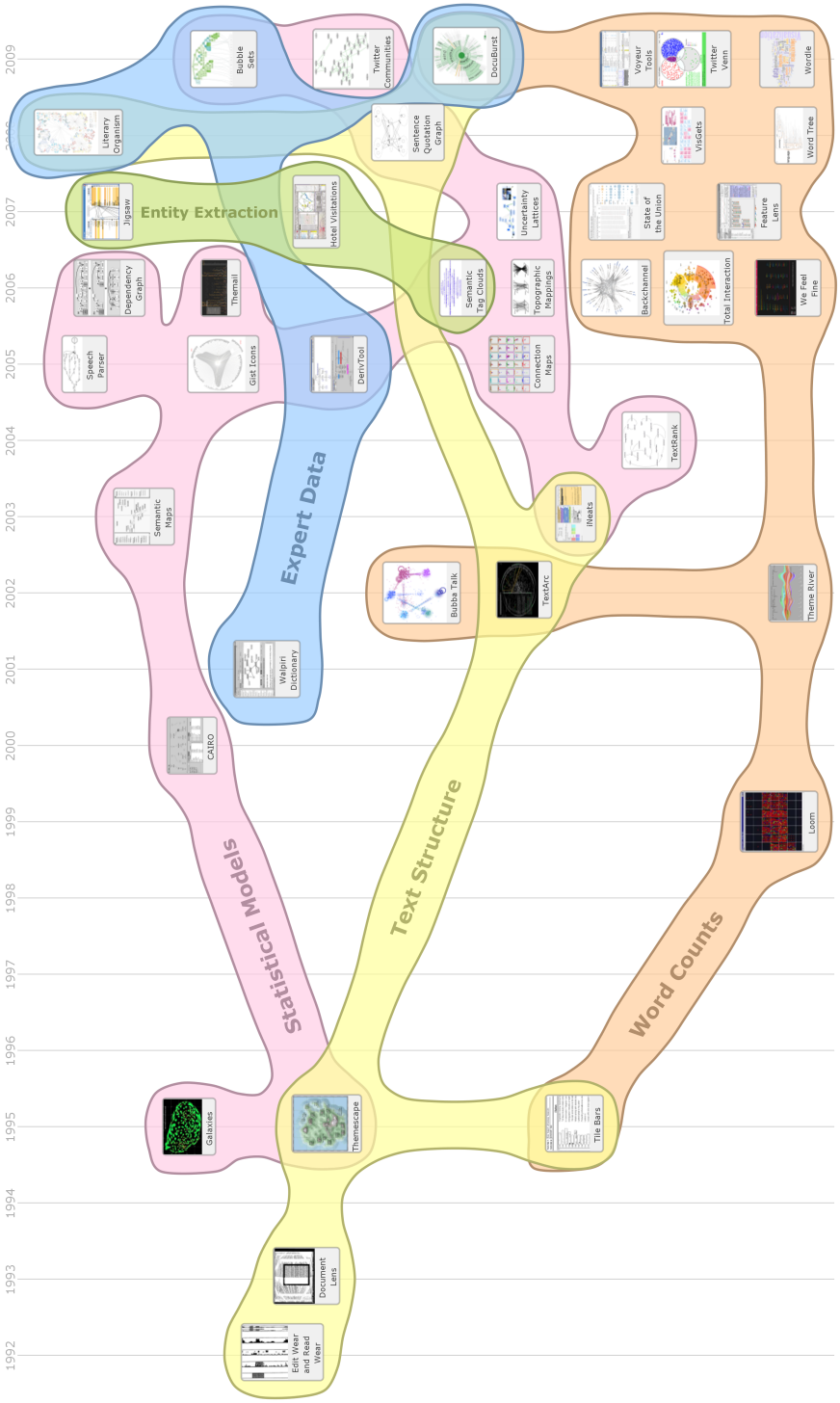


FIGURE 3.24: A Bubble Sets timeline illustrating the related work grouped by type of linguistic resource.

WORD COUNTS

How many times does a word occur (may include stemming)?

TEXT STRUCTURE

What is the pattern of sections, paragraphs, sentences?

ENTITY EXTRACTION

What are the people, places, or emotion words?

STATISTICAL METHODS

Given a trained model, what phenomena are detected?

EXPERT DATA

How does the data relate to an language-expert-created resource?

Word counting is the most commonly used type of linguistic resource. For example, *State of the Union* presents a comparative view of word counts over time (see § 3.2.5 and Figure 3.8a). Word counting can be augmented with information about the structure of the text being analyzed. For example, both *Tile Bars* (see §3.4.3 and Figure 3.21a) and *Text Arc* (see Figure 3.4.2 and Figure 3.18b) show the occurrences of words distributed from the beginning to the end of the text of interest. Rule-based entity extraction methods are also widely used, thanks in part to the popularity of online entity extraction services such as *Open Calais* (`www.opencalais.com`). Visualizations using this type of resource rely on counts of specific types of words, such as personal names, place names, and sentiment words (*e. g.*, good, hate, excite). Named entity extraction is especially common within the visual analytics community (see §3.2.9).

Moving away from the three simpler resources which rely on counting and rule-based analysis, next we have statistical methods. One may argue that a good rule-based extraction system (a very thorough set of regular expressions, for example) may be equally sophisticated as a trained statistical model. However, we separate these two methods as they require a very different set of resources and algorithms — rule-based systems may be hand-written, based on knowledge of the language. Statistical models are parameterized based on intuition about the language structure, and typically trained on many thousands of labelled examples. Linguistic visualizations produced by the CL and NLP communities often use statistical models, for example, *DerivTool* is connected to a statistical translation model (see §3.2.2 and Figure 3.5). Finally, the most linguistically sophisticated type of resource is the expert-created data. This includes linguistic-created ontologies such as *WordNet* (Miller *et al.*,

2007) and hand-annotated texts such as the document used in *Literary Organism* (see §3.2.7 and Figure 3.10b).

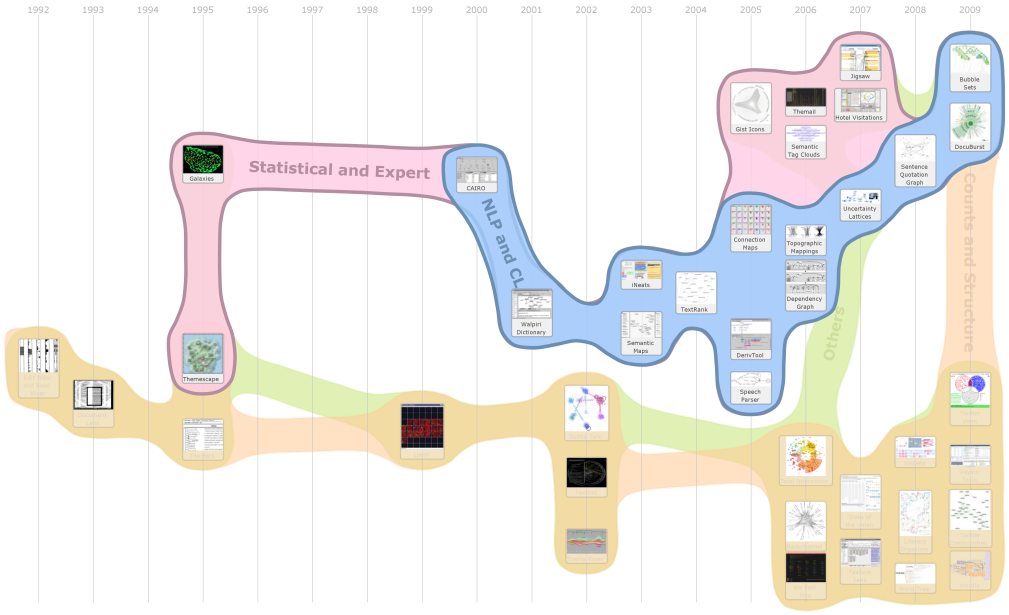
When the communities of practice are grouped on a high level into linguistic experts (CL and NLP communities of practice) and others, we can see a correlation between type of linguistic resource and community of practice (see Figure 3.25). This is expected — it makes sense that experts in linguistic algorithms and data tend to use more linguistically sophisticated resources. This clearly highlights the linguistic visualization divide.

3.7 SUMMARY

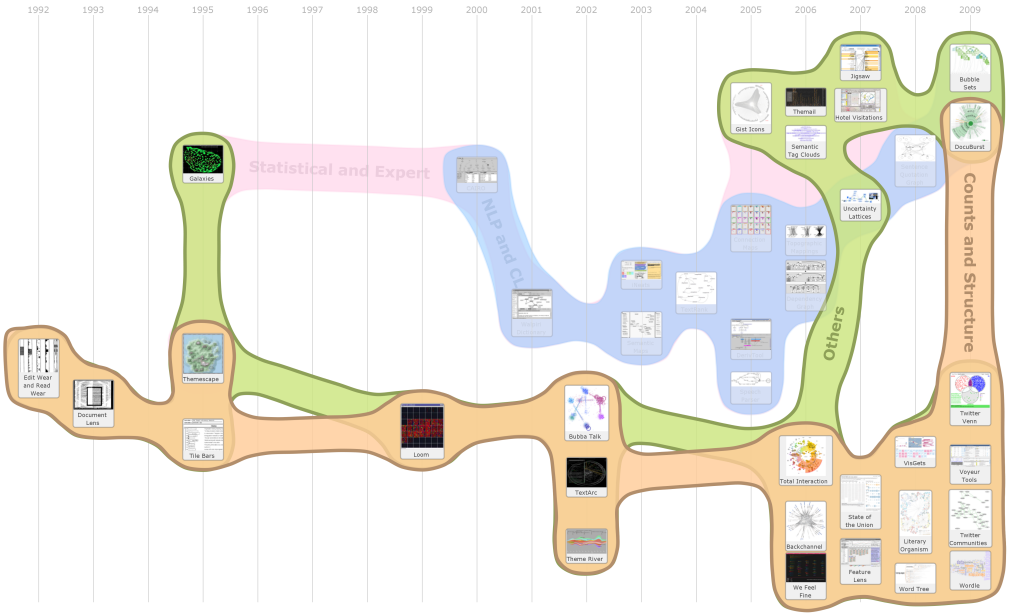
This chapter has explored the *space of linguistic visualization* using a five-way categorization based on human- and application-centred measures. Through this exploration, examples of highly interactive, but linguistically naïve (*e.g.*, *TextArc* (Paley, 2002)) as well as non-interactive and linguistically sophisticated (*e.g.*, *Semantic Maps* (Ploux and Ji, 2003)) linguistic visualization. These two ends mark the *linguistic visualization divide*. The dimensions of the space are summarized in Figure 3.26.

While we have selected five dimensions of categorization, there are other ways to divide the space, such as across the contexts of use (installation/large screen/touch table/Web deployment). The dimensions we selected to structure this chapter balance a focus on human and application-centric aspects of linguistic visualization.

In the upcoming chapters we present our five design studies of highly interactive linguistic visualization, closely coupled with NLP systems. In each chapter, we classify how the design study fits into the *space of linguistic visualization* and exemplifies a bridge across the *linguistic visualization divide*. Where appropriate, chapter-specific related work and background will be discussed to complement the projects reviewed in this chapter.



(a) The CL and NLP communities are aggregated in a single set (blue) and correlate strongly with linguistically sophisticated examples (pink).



(b) The non-linguistic communities are aggregated in a single set (green) and correlate strongly with linguistically simpler examples (orange).

FIGURE 3.25: Bubble Sets timelines illustrating the correlation between community of practice (cool colours) and type of linguistic resource (warm colours).

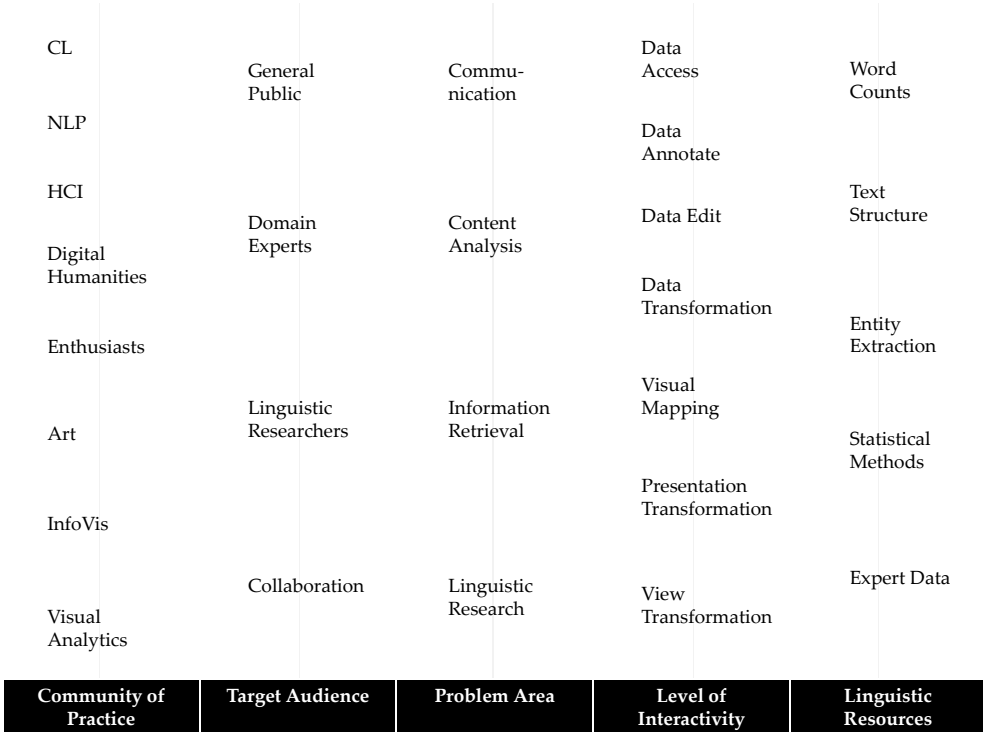


FIGURE 3.26: The dimensions and values demarcating the *space of linguistic visualization*.

Part II

VISUALIZING REAL-TIME COMMUNICATION

4

VISUALIZING UNCERTAINTY IN STATISTICAL NLP

Items of data do not supply the information necessary for decision-making. What must be seen are the relationships which emerge from consideration of the entire set of data. In decision-making the useful information is drawn from the overall relationships of the entire set.

— Jacques Bertin, 1983 (as translated from "Sémiologie Graphique", 1967)

In this chapter we present our first visualization design study, which is a generalizable decision support visualization that reveals uncertainty in lattices generated by statistical linguistic algorithms. Or, more simply, this design study presents a visualization which helps a reader of automatically translated or transcribed text to decide whether to accept the algorithm's suggested text or select an alternative from the normally hidden options. The statistical algorithms of machine translation (MT) and automatic speech recognition (ASR) underlying our visualization are designed for real-time responsiveness, making them appropriate candidates for coupling with visualization in a real-time communication scenario. Through case studies in MT and ASR we show how our visualization uses a hybrid layout along with varying transparency, colour, and size to reveal the lattice structure, expose the inherent uncertainty in statistical processing, and help people make better-informed decisions about statistically-derived outputs. By supporting exploration of the alternatives considered by these statistical algorithms, our visualization may lead to the discovery of better solutions.

As with all the research presented in this dissertation, this design study represents an initial foray into creating linguistic visualizations that bridge the information visualization (InfoVis) and natural language processing (NLP) communities of practice. The design is targeted at a general audience of people who send and receive instant messages or use speech transcription software — we assume no linguistic expertise.

The problem we consider in this visualization is how to use visuals to improve machine translation results by bringing "the human into the loop". We strive for "walk-up-and-use" usability through careful selection of

visual encodings based on the well-known properties of human perception (see Chapter 2), as well as drawing on common metaphors such as 'fuzzy' meaning 'uncertain'. While lattice structures are used as black boxes in many processing systems, we know of no other visualization to support their use in "human-in-the-loop" decision making.

The levels-of-interaction of this design study are at the presentation level and the data access/edit level (see §4.5.4). Changes in presentation such as varying the use of border transparency, edge colour and photo size are supported. Interaction is also provided through two data change operations which are persisted in the interaction history (chat log): input new data and select an alternative best path. Note that unlike most information visualizations, this design study does not allow for view changes — the layout and zoom factor are automatically optimized for reading the representation.

4.1 MOTIVATION

Many NLP applications are statistically based (*e.g.*, Jelinek, 1998; Knight, 1999; Koehn, 2004). Their outputs represent a "best guess" by the algorithm, given some training data, parameter settings, and input. These best-guess outputs come from a very large collection of possibilities, each ranked with a score. However, these systems present their result in a black-box fashion, showing only a single response. Since no details about probabilities, uncertainties, or the workings of the algorithms are provided, it is easy to misconstrue the output as having a low uncertainty. This lack of detail deprives us of the context necessary to make well-informed decisions based on the reliability of that output.

4.1.1 *Uncovering Hidden Information in Word Lattices*

Applications such as MT and ASR typically present a best-guess about the appropriate output as their result, with apparent complete confidence. Internally these algorithms often generate many possible alternatives, all with assessed degrees of uncertainty. These alternatives are stored in lattice graphs, which are the underlying data structures in many statistical processing systems, including many NLP applications. Lattices compactly represent multiple possible outputs and are usually hidden from view.

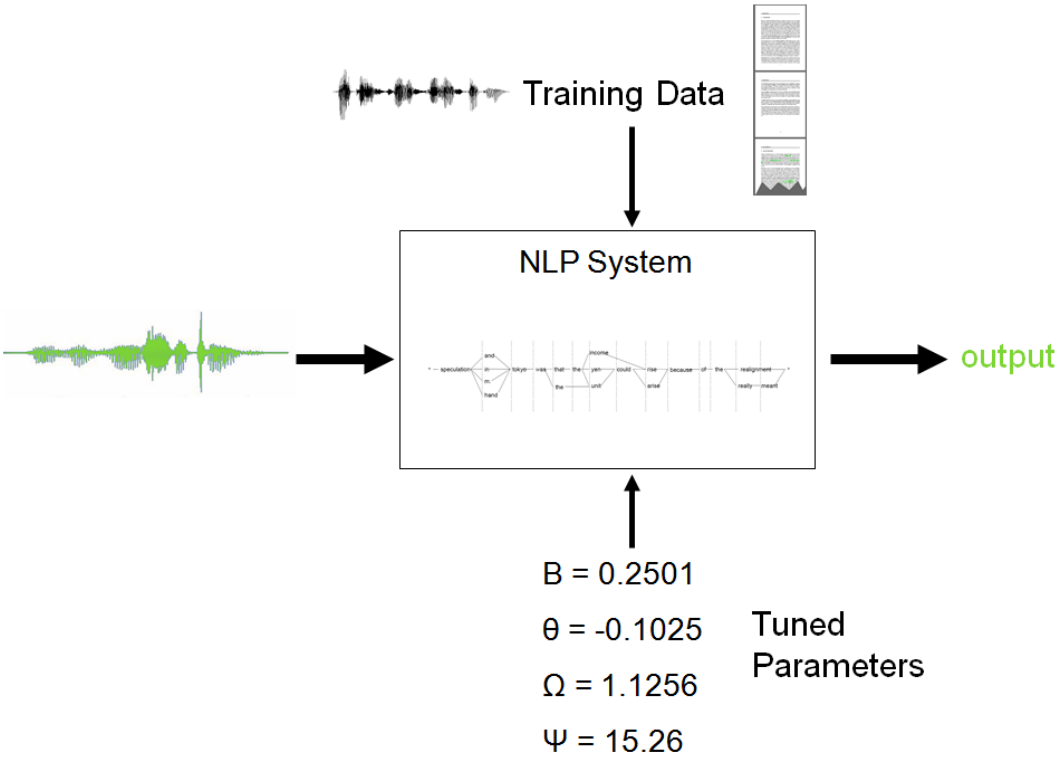
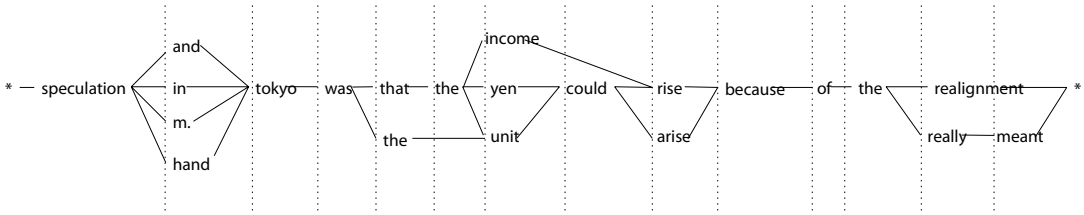


FIGURE 4.1: Statistical NLP systems often contain word lattices as an internal data structure, used in a "black box" fashion, with a single best output provided to the viewer. This design study seeks to expose the lattices using intuitive visualization techniques to give additional control to the user of the NLP system.

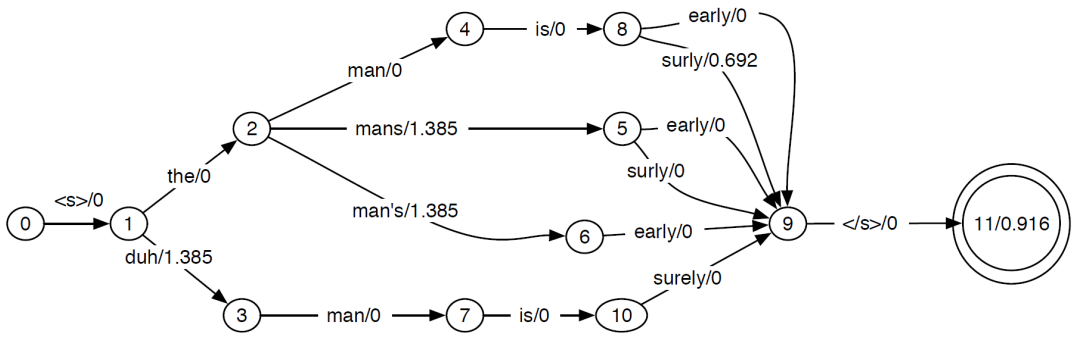
Figure 4.1 illustrates the centrality of word lattices to a typical statistical ASR system.

While general graphs and some subsets of graphs such as trees have received considerable visualization attention, other important subsets such as lattices have been largely ignored. Lattice graphs are used as the underlying data structures in many statistical processing systems and serve well in holding the possible ranked alternative solutions (Jurafsky and Martin, 2000).

As evidence of the linguistic visualization divide, we can find representations of lattice graphs in published papers from the NLP community. These information graphics are used to communicate data structures and algorithm details to an audience of linguistic researchers. They are not available real-time from working systems, they are not interactive, and



(a) A simplified word lattice. Dotted lines represent time alignments (Collins, 2004).



(b) A sample word lattice transformed into a weighted finite state machine (Hall, 2005).

FIGURE 4.2: Depictions of word lattices from NLP research. *Reprinted with permission.*

do not visually encode the wealth of information, such as uncertainty scores, available from the MT model. In our own previous work, we have published manually-created sketches of word lattices for the purposes of illustrating the structure of the data, which was used as input for a parsing system (see Figure 4.2a) (Collins, 2004). Similarly, Hall (2005) published several manually-drawn lattice representations for the purpose of explaining how the data structure is used in MT and as an input to a parsing system. Figure 4.2b shows an example of a word lattice represented as a weighted finite state machine. On the other side of the divide, the graph drawing community has been very interested in tools to interactively construct and display Galois *formal concept lattices* and *Hasse diagrams*, neither of which are visualized using methods amenable to our task (see Figure 4.3).

The design presented in the following sections draws on the aligned spatialization of Figure 4.2a and the inclusion of uncertainty scores as in Figure 4.2b. This is automatically drawn using a combination of grid and force-based techniques to create a layout that focuses on multiple

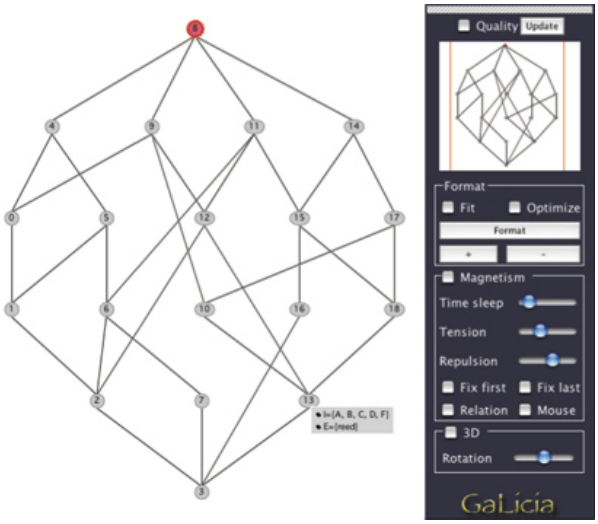


FIGURE 4.3: The Galicia tool for constructing and working with concept lattices (without any connection to language) (Valtchev *et al.*, 2003). *Reprinted with permission.*

encodings of the uncertainties using position, transparency, and colour (see Figure 4.4). These statistically-derived lattices are amenable to visualization since the uncertainties are locally constrained. The uncertainty encodings presented here are readily applicable to representations of uncertainty in general graphs.

4.2 REVEALING UNCERTAINTY TO SUPPORT DECISION-MAKING

Understanding about the human reasoning process informs us that, while not idealized Bayesian decision-makers, people do make decisions based on their analysis of the objective context of the problem and subjective probabilities informed by their personal body of knowledge (Cohen, 1979). For example, in the context of a natural language system such as MT, a person makes a decision about the reasonableness of the output based on their prior knowledge of likely constructs in the language. Based on Cohen's review of research on reasoning, we work with the assertion that the quality of the decision about whether to accept the algorithm's best guess can be enhanced by knowing the uncertainty inherent in the solution. That is, providing easy access to the objective context will enable people to make better decisions. Since the effort a person will want to expend in making a decision is proportional to the perceived importance

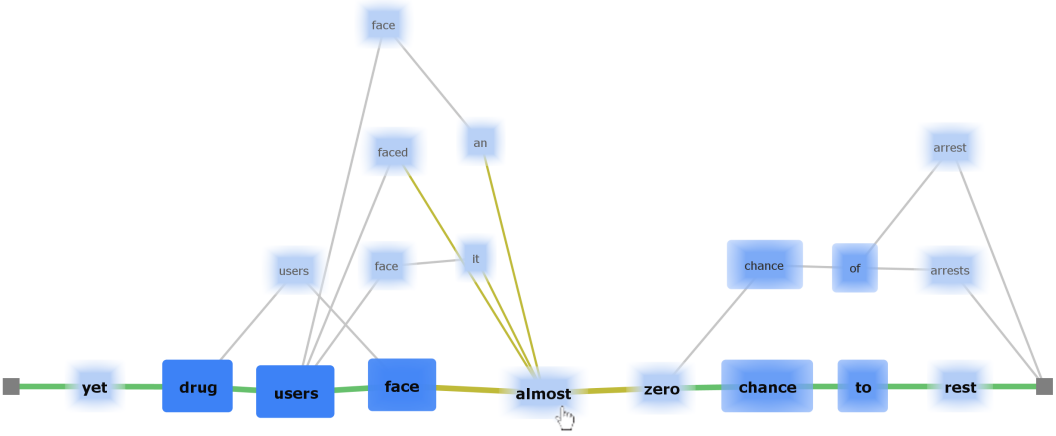


FIGURE 4.4: A speech recognition lattice for which the statistically-identified best path is not the best transcription. The best path according to the model is shown in green. Node uncertainty is represented by position, fill hue, and edge transparency gradient. Edges attached to the node under the mouse pointer are coloured gold.

of that decision, the algorithm's best guess should remain obvious while providing visual access to ranked probabilities. For instance, a person may accept a confusing translation in casual conversation in an Internet chat-room, but would reject the same problematic translation in a multi-lingual business negotiation.

There are many aspects of language modelling that statistical processing has yet to master — for instance, an output of speech recognition occurring in the corpus we use is, "the prisoners resisted a rest." Without our visualization one would not know that "the prisoners resisted arrest" was the second-highest scored hypothesis. While any native speaker would guess the correct reading of the phrase, presenting it visually in parallel with the algorithm's best guess removes the puzzle aspect for a native speaker but provides a learner with the needed support. By revealing alternative hypotheses considered by the algorithm, and the uncertainties associated with each, our visualization shows promise for facilitating the process of recognizing and correcting of errors.

4.3 BACKGROUND

As information visualization as a field has matured, focus on visualizing uncertainty in a dataset in conjunction with the absolute data values has increased (Johnson and Sanderson, 2003). Amar and Stasko (2004) call for

bridging of analytic gaps between data representation and analysis, and one technique they suggest is to expose "black box" processes through visualization of meta-data such as uncertainty. Examples from the literature that are relevant to our approach include using line thickness and transparency to represent uncertainty in architectural drawings of ancient structures (Strothotte *et al.*, 1999) and using iso-surfaces and motion blur to represent uncertainty in molecular (node-link) diagrams (Rheingans and Joshi, 1999). Zuk and Carpendale (2006) present a theoretical analysis of these and other uncertainty visualizations in which they summarize the significant theories of Bertin (1983), Tufte (2001), and Ware (2004) (see Chapter 2) and apply them as heuristics for evaluation of visualizations of uncertainty. We draw upon their analysis for design guidance. We will reflect more upon our design choices based on these theories in the following sections, after a brief review of lattice graphs and lattice graph visualization.

Formally, a partially ordered set L is a lattice if, for all elements x and y of L, the set $\{x, y\}$ has both a least upper bound in L and a greatest lower bound in L. Lattices are used in formal concept analysis (Galois lattices), and have been previously visualized using simple force-directed layouts (Valtchev *et al.*, 2003). Lattice drawing has also been of interest to the universal algebra and graph drawing communities, where the focus has been on drawing Hasse diagrams: the edges must be straight lines and the vertical coordinate must agree with the order. Reducing edge crossings has been a primary concern (Freese, 2004). Our goal differs in that we are not focused on understanding the particular formal structure of the lattice, but rather using that structure to support understanding of the data and the uncertainty represented by it.

The 'lattices' in statistical processing do not meet all conditions of this formal definition. Intuitively, we can imagine a lattice in this work as a partial order with a unique beginning and end. Seen as a graph, for every node in a lattice there exists a path from the beginning to the end which passes through that node. To our knowledge, neither lattices for statistical processing nor uncertainty within lattices have been previously visualized except for use in explanatory information graphics (see Figure 4.2a, 4.2b).

4.4 DATA

The lattices generated by statistical processing are collections of nodes and edges, each associated with a data value (for example, a word) and a score (for example, an uncertainty). A lattice is generated as a representation of the solution space for a given problem; any path from beginning to end through the lattice represents an hypothesis about the solution. However, the lattice may or may not contain the true solution. For example, a speech recognition lattice contains all the potential transcriptions considered by the algorithm. It may not, in fact, contain the correct transcription. Each lattice has a *best path* through it, based on node scores as well as a *true best path* which, while it may not have the best node scores, best matches the true solution. Our goal is to use visualization to provide an opportunity for people to combine the scores with their world knowledge to discover the *true best path* or to reject the entire lattice.

In a lattice generated by a statistical process there may exist a unique start and end node, representing the beginning and end of all possible solutions. If such endpoint nodes do not exist, we create them, extending edges from all starting and ending nodes to the new endpoints. Unique endpoints provide for an easy to locate visual entry-point to reading the lattice. Our visualization algorithm reads lattices from the statistical processing (source) algorithm using either an interface customized to the application, or the HTK Standard Lattice Format (SLF) (`http://htk.eng.cam.ac.uk`). In our current work, we only use labels and scores on nodes. We convert lattices with edge probabilities to have posterior probabilities on the nodes using the SRILM lattice toolkit (Stolcke, 2002). Finally, we retrieve the best path through the lattice, according to node scores, either directly from the source algorithm or using the SRILM lattice toolkit.

When we discuss the uncertainty of lattice nodes, we do not strictly mean uncertainty, as might be quantified by an entropic measure, for example, but rather a more application-specific property that emerges from the lattice which reflects the probability that the node is part of the true best path. In particular, node scores are generally relative confidence scores, not true probabilities, and the presentation of several alternatives at any slice in the lattice is more an indication of the number of plausible solutions, rather than of a small margin of preference among those alternatives. The score of a node, nevertheless, can be interpreted as a measure

of certainty that the node is the correct choice for its span and appears in the best path.

4.5 LATTICE UNCERTAINTY VISUALIZATION DESIGN

Traditional statistical processing systems use a large corpus of data to quickly produce a single hypothesis, drawing on a computer's strength in dealing with large amounts of data with the goal of quickly solving a problem. However, if one is presented with the best result of a statistical process, but the quality is so poor it is not useful, then the original goal of providing convenience is not met. Building on the generalization of human-computer optimization by Scott *et al.* (2002), we hypothesize that by including a human "in-the-loop" we can leverage the intelligence of the human and the processing power of the computer to quickly solve the same problems with better solutions.

4.5.1 *Design Goals*

To meet this goal, we identified several constraints to guide our design process:

- ensure easy readability of paths in the lattice;
- provide an intuitive visual mapping of uncertainty within the lattice which supports the ordering of nodes;
- provide for visual pop-out of nodes of high certainty and nodes in the optimal path identified by the algorithm;
- provide alternative representations of the data to clarify meaning, where possible;
- in most cases, require no interaction;
- where interaction is necessary (providing detail in context and manipulation of best-path tracing), it should be lightweight and easy to learn.

In order to ground our design in an understanding of human perceptual capabilities, we investigated the properties of visual variables (§ 2.3.3), leading us to select those that allow for high-speed estimation (Table 2.3.3) to convey relevance, and that provide an ordered reading to convey uncertainty. From this, we chose edge sharpness, hue, and position

to make nodes of high confidence stand out, and uncertain nodes less visually prevalent — these visual encoding choices are informed by Zuk and Carpendale's 2006 theoretical analysis of uncertainty visualizations. Also, since value, size, position, and transparency are ordered (values can be visually sorted), we used these to encode uncertainty to allow for comparison of the relative scores between nodes.

Data and uncertainty about lattices is localized to individual nodes, allowing us to take advantage of the concept of small multiples (repeated graphical elements using the same encoding): once the mapping of uncertainty to node appearance is understood, this can be used to interpret all the nodes in the graph (Tufte, 2001).

4.5.2 *Layout*

In the graph drawing community, where the lattices are usually representations of an algebra, the convention is to draw the order vertically, from bottom to top (Freese, 2004). However, in the languages our visualization is designed to support, text-based reading occurs left-to-right. Additionally, temporal flow (as in the flow of speech signals) is usually thought of as occurring left-to-right. So, to support our design goal of easy readability, we align our visualization horizontally to allow for more natural left-to-right tracing of paths.

Our layout algorithm is a hybrid of grid and force-based layouts. Initially, the lattice graph is laid out on a grid, anchored by endpoints which are positioned according to the length of the algorithmic best path through the lattice. This avoids large gaps or significant overlaps. Horizontal node positioning is derived from the node order in the lattice from beginning to end.

Vertical position is assigned to separate nodes covering the same span, ordered by increasing uncertainty bottom to top. Because the algorithmic best path is of most interest, we place it along the bottom, regardless of the individual node scores. This anchors the visualization in the algorithm's best-guess solution and facilitates easy reading of it (see Figure 4.5a). Position, the strongest visual variable, according to Bertin (1983), ensures that the least important nodes (highest uncertainty) appear furthest from central visual focus along the bottom. In other words, we assign *spatial rights* to nodes most likely to be correct.

The grid layout can sometimes result in overlaps for nodes with lengthy labels and for larger lattices. We automatically zoom out the display to attempt to fit the entire lattice without overlap, but must limit the scale factor to ensure label legibility. To reduce overlap, we adjust the layout using a force-directed energy minimization. An unconstrained force-directed layout alone would create an unpredictable and unordered layout (see Figure 4.5b). Thus, nodes are anchored by invisible springs to their grid positions, and to each other by springs represented visually as graph edges. Repellent forces are applied between nodes to reduce overlap and the energy minimization is run for several seconds to stabilize the layout. This hybrid layout allows any overlapping nodes to separate while not moving far from their grid-determined position, balancing the need to keep nodes in the rigid layout for easy left-to-right reading and the demand that nodes do not overlap (see Figure 4.5c).

4.5.3 *Uncertainty Encoding*

Uncertainty in the lattice is foremost visualized through the presence of alternative paths through the lattice: more paths can indicate greater uncertainty, depending on the relative scores for the nodes in each path. Uncertainty scores are used to colour the nodes using a range from saturated blue to desaturated gray. However, continuous colour scales generally should be avoided for numerical data (colour perception varies due to several factors, including the size of items) (Ware, 2004). To compensate for this, we redundantly encode the scores in the node border using size and transparency. Hue, border size, and outer edge transparency are all linearly related to the uncertainty score on each node, with maximum and minimum visual values clamped to the maximum and minimum uncertainty values within each lattice. This is a consequence of our case study data in which node uncertainty is only comparable within, not across, lattices. Note that if the source data had a consistent and comparable notion of uncertainty scores across lattices, then the visual encoding would be better as linearly relative to a global maximum and minimum.

We present two alternatives for encoding, each with its own advantages (see Figure 4.6). In the "bubble border" view, the node border varies from a tight solid blue, indicating high confidence, to a transparent, wide, gray

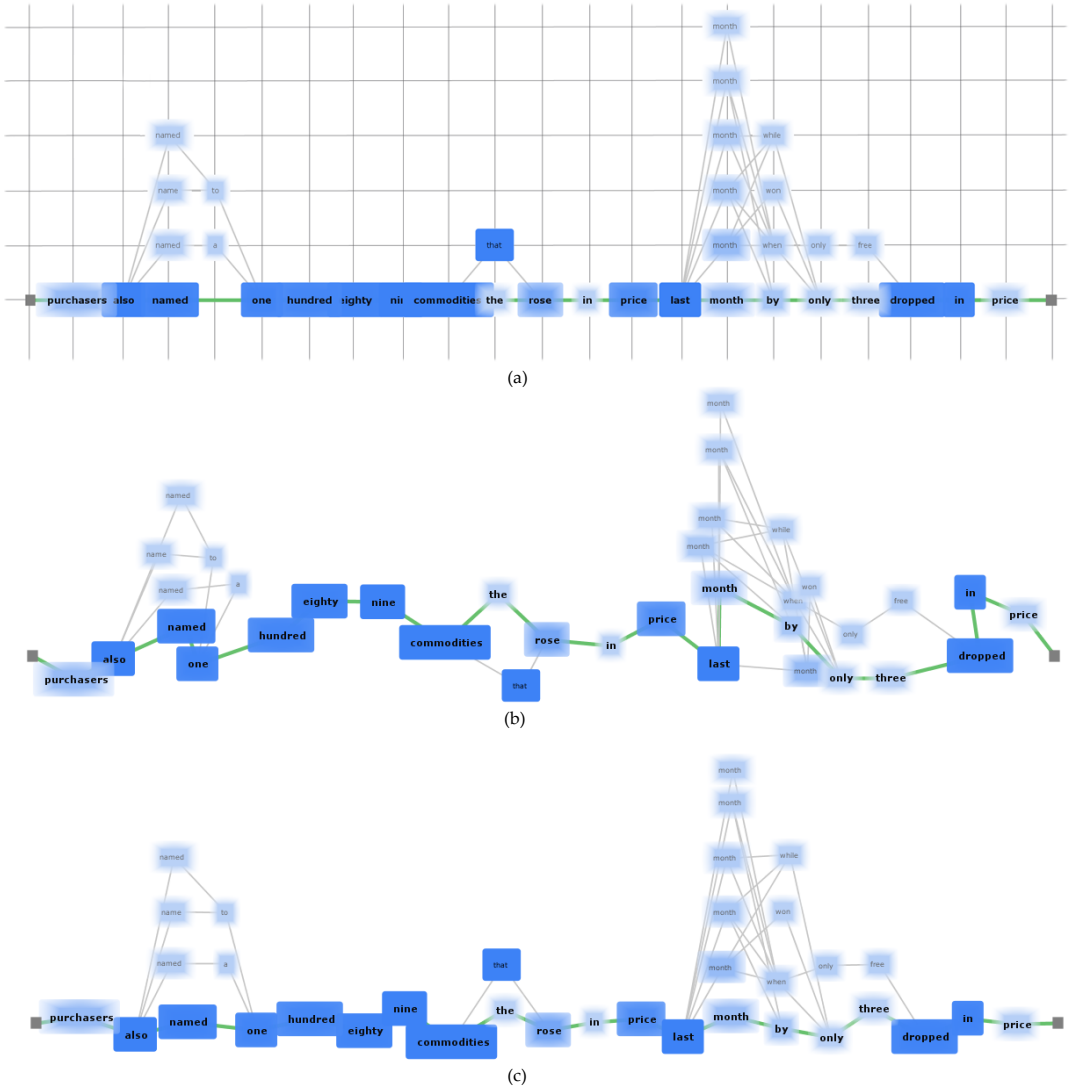


FIGURE 4.5: Layout construction: (a) rigid grid-based lattice, (b) force-directed layout, (c) hybrid layout. The hybrid layout provides the regularity benefit of the grid-based layout and the overlap avoidance of the force-directed layout.

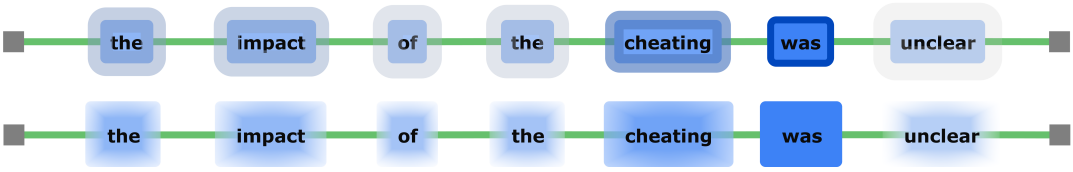


FIGURE 4.6: Two alternative encodings of uncertainty. The top, "bubble border", uses border thickness, hue, and transparency to convey uncertainty: tight, solid blue borders have a higher certainty. The bottom, "gradient border", uses blurring of the edges through a gradient and variable transparency: more solid borders have higher certainty. Both variations also use hue variation from bright blue (high certainty) to gray-blue (low certainty).

border, indicating uncertainty. Large, semi-transparent borders lead to an intuitive reading of uncertainty.

In the "gradient border" view, the node border varies from a crisp edge to a gradient leading to complete blending with the background. The gradient border is achieved through a linear blending of full opacity at the node center to variable transparency at the outer edge. This effect simulates semantic depth-of-field (Kosara *et al.*, 2001) in which items with crisp focus pop-out (a 'preattentive' effect (Table 2.3.3)). Even though the gradient fill on the nodes in this view does not overlap the text label, in informal testing we found that the blur effect seemed to make the labels more difficult to read. So, while the gradient border may be more intuitive and lead to a more immediate reading, the bubble border may, in the end, be a more usable encoding.

In both cases, the use of transparency is supported by visualization theory: transparency blends the visual variables of value and colour in a redundant encoding from which an ordered reading is possible (Bertin, 1983). These techniques satisfy our goal: to coarsely and quickly indicate relative uncertainty without providing specifics on the scores of each node. In fact, the precise numbers are often not very meaningful: they result from the settings of many variable parameters in the model which generated the lattice and are generally only comparative within a particular lattice.

4.5.4 *Interaction*

Simple interaction techniques are provided: when hovering over a node, its edges are highlighted in gold to disambiguate edge crossings (see Fig-

FIGURE 4.7: Edges connecting a node are highlighted on mouse over to clarify edge crossings.

ure 4.7). Nodes can also be dragged to further clarify edge attachment, returning to their original location when released. By right-clicking nodes, one can remove and add nodes to the green-edged best path, thereby using their knowledge of the context (for example, their prior linguistic knowledge) to reassign the best path (see Figure 4.8). Where others have used iterative feedback to recompute a new best path through a (hidden) lattice based on human feedback (*e. g.*, Liu and Soong, 2006), we provide complete control to the human decision maker. For our interface, an iterative process is unnecessary as the entire lattice is visible. Furthermore, iterative interaction would violate our minimal interaction design constraint. In the case studies to follow, we will explore how this functionality can be applied in real implementations.

4.6 CASE STUDY: MACHINE TRANSLATION

Machine translation offers much promise for improving workplace communication among colleagues situated in offices in different parts of the world. Many corporations use instant messaging chat as a means of facilitating communication, however current translation quality is too low to feasibly use it in a critical setting.

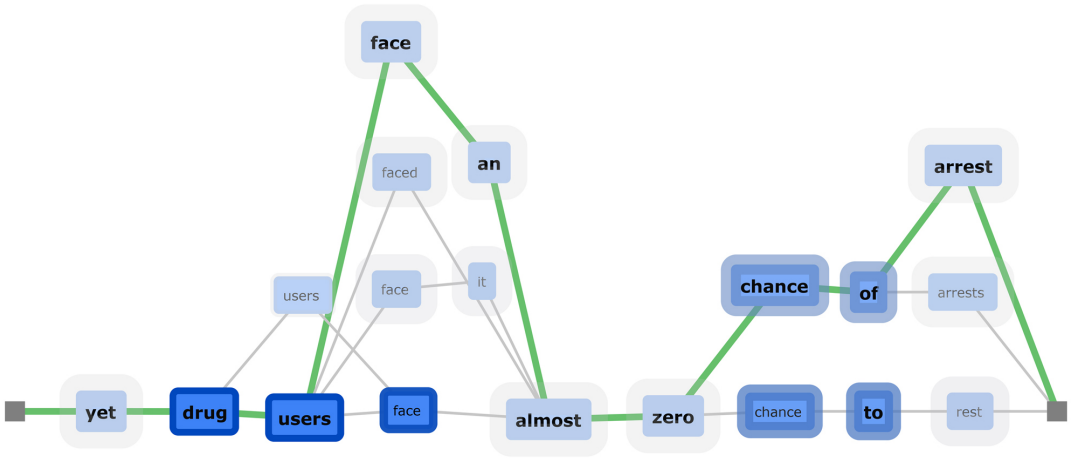


FIGURE 4.8: Lattice from Figure 4.4 with best path corrected.

In this case study we present a prototype visualization system for instant messaging conversations which uses our lattice uncertainty visualization to reveal the uncertainty in the translation and provide alternative translations when available (see Figure 4.9).

Several visualizations of different aspects of instant messaging conversations have been reported, for example, visualizations have been created of chat histories (Tat and Carpendale, 2006) and real-time conversations (*e. g.*, Tat and Carpendale, 2002; Viégas and Donath, 1999). However, to our knowledge, no one has investigated visualization as a support for a cross-language chat.

Despite evidence that social spaces in the Internet are multilingual in nature, these spaces still lack rich cross-linguistic communication (Herring *et al.*, 2007) and little research has been directed toward supporting cross-lingual chat. Recent studies on cross-lingual instant messaging chat in distributed workplaces show that poor translation quality negatively affects conversations (Yamashita and Ishida, 2006). To our knowledge, only a few commercial cross-lingual chat applications (*e. g.*, `http://www.chattranslator.com`) exist and they only present the best-path solution to the viewer.

Visualization tools for translation researchers (*e. g.*, Albrecht *et al.*, 2009; DeNeefe *et al.*, 2005; Smith and Jahr, 2000) are related to our visualization in that they provides a means for exploring alternative translations. However, where we focus on providing a visual means to understand translation uncertainty, targeted at end-users of translation systems, the

other visualizations are tailored for examining and evaluating specific word correspondences between languages. We will revisit these systems in Chapter 7, which is more closely related.

4.6.1 *Translation Architecture*

We chose to work with instant messages as the data for uncertainty visualization in translation because they offer several advantages for this work. They tend to be short, keeping translation time low and providing an appropriate amount of data for a small-scale visualization. The result should be a manageable number of alternate translations for chat participants to explore. We developed a bidirectional instant messaging client which performs translation on messages it receives using a beam search decoder for statistical phrase-based translation models. We trained the decoder, Phramer (Olteanu, 2006) (an open-source implementation of (Koehn, 2004)), on the English, Spanish, French, and German portions of the Europarl corpus (approximately 1M sentences in each language) (Koehn, 2003). The phrase-based translation is supported by a trigram language model trained on the same data (Stolcke, 2002). The translation algorithm evaluates the input data and creates a set of translation hypotheses, assigning confidence scores to each word and phrase based on occurrences in the corpus. The best path through the lattice, according to the scores, is labelled by the translation system. Using this data, we create a compact lattice populated with all alternate translations which have a score within a preset threshold of the best score. This graph, complete with scores for each node, is then used as the lattice for visualization.

4.6.2 *Interface*

In following with norms of instant messaging client design, we maintain a chat transcript: the green-edged best path is recorded to the chat history when the next message is received. However, it often occurs that a node along this path has a low confidence score (high uncertainty). The conversant can explore alternative translations for this span of the sentence, or, if no reasonable alternatives exist, use the chat to request clarification from the author of the original message. When out-of-vocabulary words are encountered, or the translation uncertainty is particularly high, photos are retrieved from *Flickr* (`http://www.flickr.com`) using the original

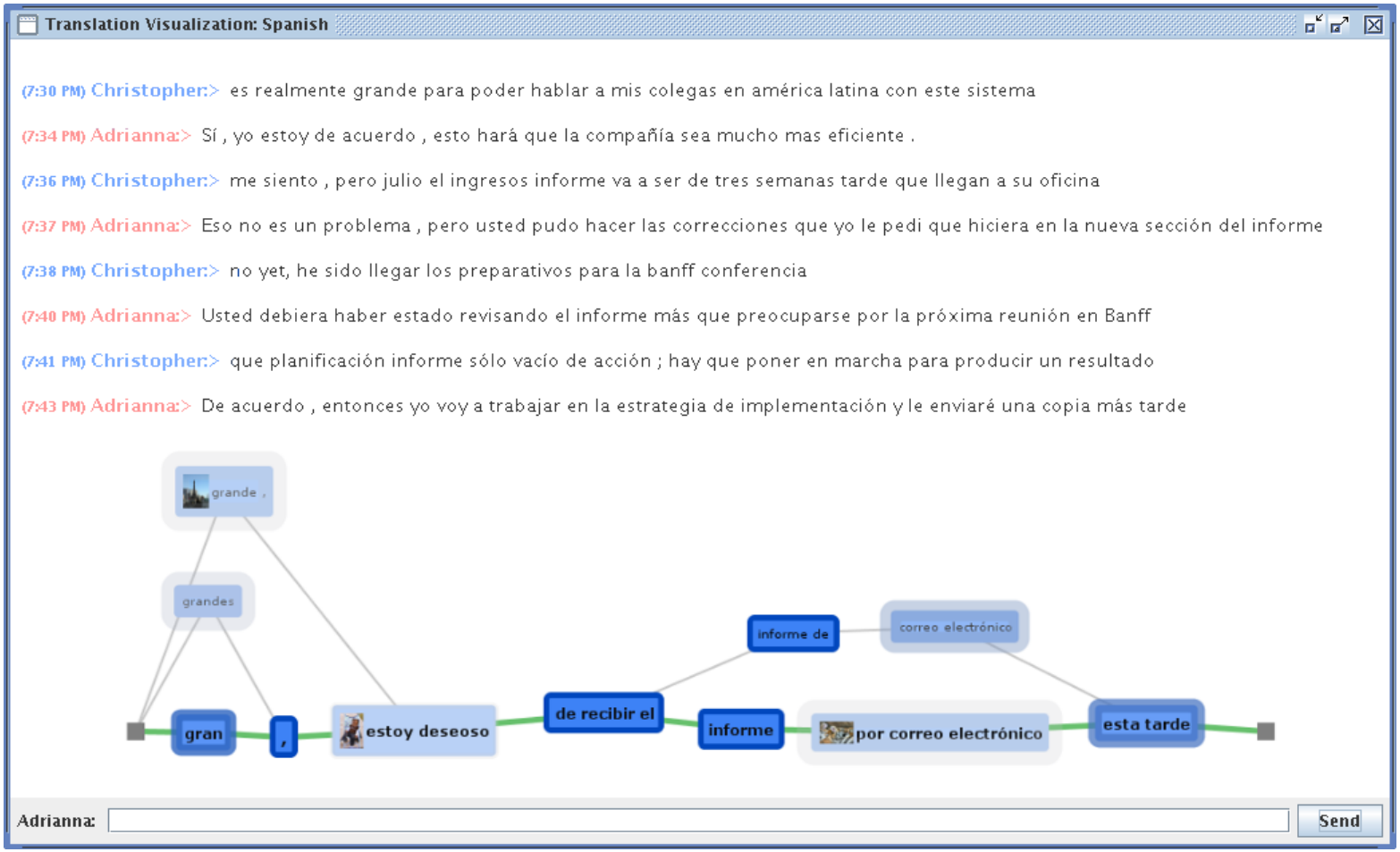


FIGURE 4.9: Translation uncertainty visualization in a chat interface: upper half records chat history in the language of the local conversant, lower half displays the visualization of the most recently received message. Translations of low confidence are also augmented with representative pictures collected from the Internet.

(untranslated) words as a search query. In some cases, images may easily clarify the intended meaning (see Figure 4.10).

The main interaction is through the chat message box and reading the data presented in the visualization. To facilitate accurate chat logging, the ability to toggle node inclusion in the green "best path" is provided. In this way, alternate translations can be selected and recorded in the log instead. Selecting a photo node enlarges it, revealing a set of four images about that node.

4.6.3 *Discussion*

This chat system was designed for two participants in online meeting, neither of whom speaks the other's language. Through our instant messaging system, they converse, in some cases exploring the lattice uncertainty visualization structure for clarification of a poor translation, and in other cases rejecting the entire translation as too low quality based on the node uncertainties.

This visualization and chat system was demonstrated at *CSCW 2006* (Collins and Penn, 2006). Informal participant feedback indicated an interest in multi-lingual chat in general, and in the visualization of uncertainty. Participants indicated they would like to try the system for a longer period of time, in particular they liked the inclusion of photos on untranslatable nodes. From using the visualization, we notice that for English translated to French or Spanish, many of the lattices have ambiguities on single words and short phrases, whereas for English to German there are longer segments of ambiguity, likely due to the freer word order of German.

4.7 CASE STUDY: AUTOMATED SPEECH RECOGNITION

Automated speech recognition is another application area where lattices are commonly used in processing but only the best solution is reported. The selection of the best path is dependent on the quality of the speech input signal, the acoustic model, and the language model. With many places to go wrong, speech recognition often produces incorrect results.

There have been investigations into using lattices to suggest alternative translations in drop-down lists and in multi-modal interfaces, including handwriting recognition (Suhm *et al.*, 2001), but generally people remain

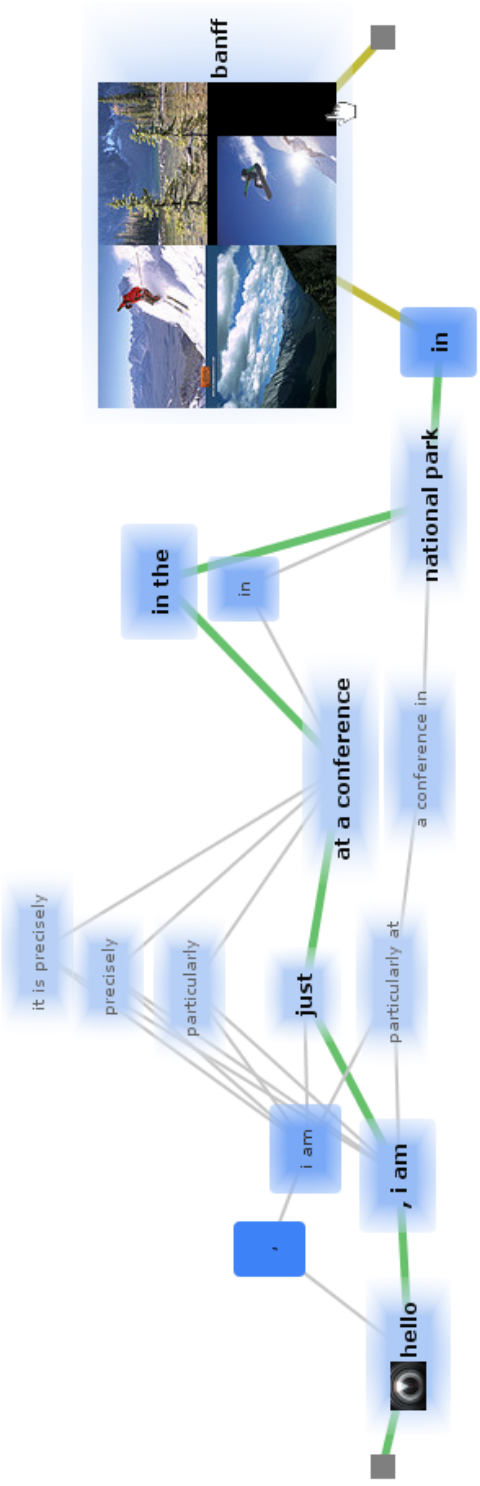


FIGURE 4.10: Translation lattice for the German sentence, "Hallo, ich bin gerade auf einer Konferenz im Nationalpark in Banff." The statistically-identified best path (along the bottom) was incorrect and has been repaired. Photo nodes provide an alternative representation for words not in the translation vocabulary. Mouse-over expands the node and reveals four photos, while other nodes move away to avoid occlusion.

dissatisfied with these interfaces. Kemp and Schaaf (1997) report on a text-based tagging system which labels each word of speech recognition output with a normalized measure-of-confidence score. However, in their work, alternative hypotheses are not provided. In all cases, the lattice structure remains hidden from view. Although much attention has been given to supporting correction of transcription errors, we know of none that use the lattice and its scores directly in "human-in-the-loop" interaction.

4.7.1 *Recognition Architecture*

Algorithms for automated speech recognition are generally arranged as a pipeline of data transformations. For our purposes, we can think of this pipeline as a three step process:

1. an acoustic model takes a digitized speech signal and creates a word lattice with scores,
2. a language model re-scores the lattice based on probabilities of words occurring in sequence,
3. the best path through the lattice based on the acoustic and language model scores is output.

The *NIST '93 HUB-1* collection of word lattices represents data captured from this process after step 2. This collection of 213 lattices comes from high-quality recordings of 123 speakers reading excerpts of the *Wall Street Journal*. Note that in the HUB-1 collection, some node labels may be repeated, indicating multiple possibilities arising from uncertainty about the start time or length of the word in the speech signal. The lattices include acoustic and language model scores along the edges. We used the SRILM lattice toolkit to calculate scores for the nodes and prune the lattices to contain at most the 50 best unique paths. We also eliminate null nodes (silences) and nodes with scores below 0.01% of the best scoring node. While our visualization is decoupled from the actual speech signal, it could easily be connected to the speech recognition pipeline directly.

4.7.2 *Discussion*

Examples of visualization of the HUB-1 lattices appear in Figures 4.4–4.8, and there are many examples from this case study for which the best path chosen using the node scores is not the true best path in the lattice.

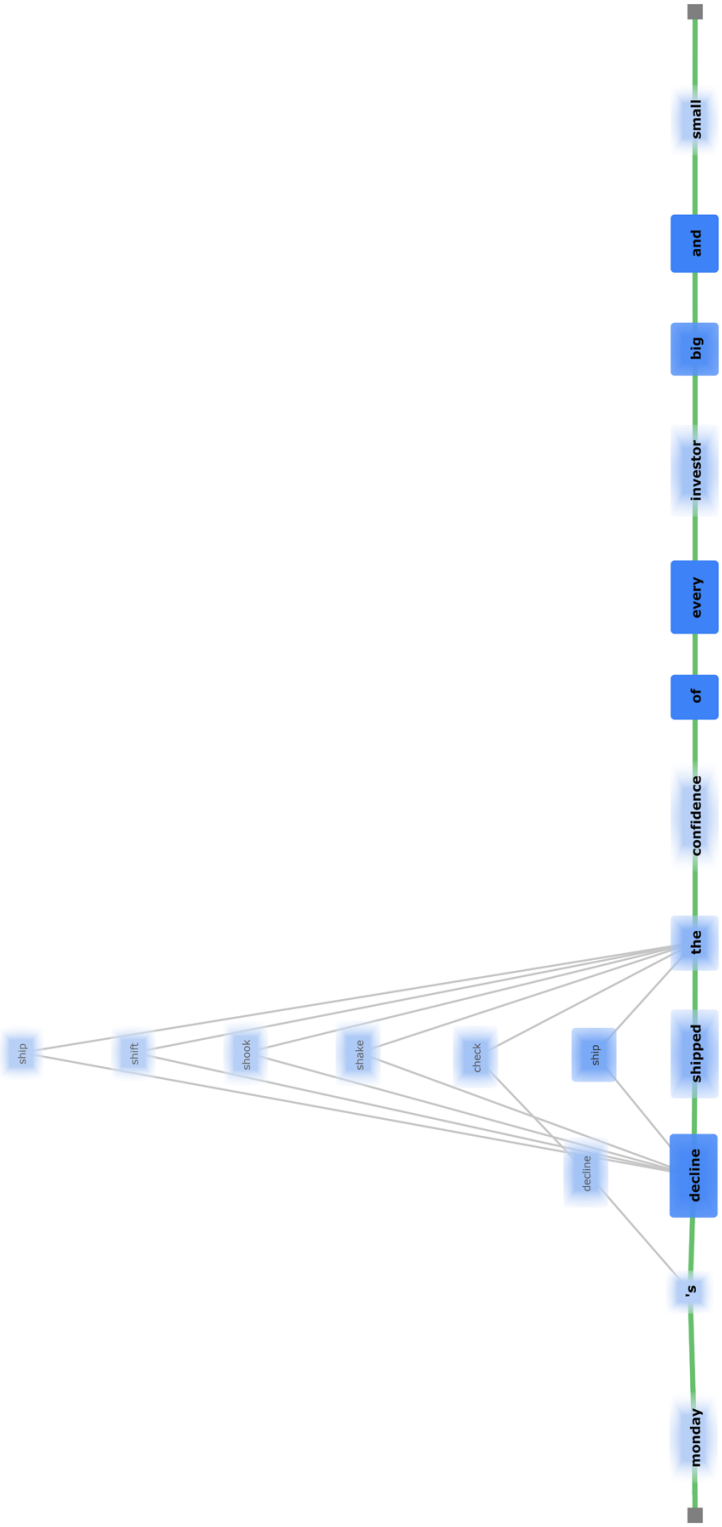


FIGURE 4.11: Speech recognition lattices often have localized areas of high uncertainty.

In informal testing, it seemed that in many cases, the correct path was obvious upon reading the optional nodes for a particular span — only one path made sense. Through using the visualization, we discovered that the speech lattices seem to generally have a different structure than translation lattices: where ambiguity in translation often presents an alternative or two for a span of several nodes, speech recognition lattices show highly localized ambiguity (see Figure 4.11). This stems from the difficulty of acoustic models for speech recognition to recognize short words; a short duration and low signal amplitude lead to elevated uncertainty. By coupling our visualization of uncertainty with human linguistic knowledge, it is possible to make better informed decisions about the quality of a transcription, and to correct errors by selecting a new path in the lattice. In this way our visualization could support real time editing of speech transcripts on a sentence-by-sentence basis.

4.8 SUMMARY

This chapter presented a generalizable visualization for uncertainty in lattices generated by statistical processing. The techniques for visually encoding uncertainty may be applicable to other node-link structures, such as Hidden Markov Model trellises, probabilistic finite state automata, and general graphs. As our visualization reveals the search space considered by common statistical algorithms in areas such as NLP, it could be useful as a teaching tool.

Figure 4.12 summarizes how this design study fits into the *space of linguistic visualization*. This study represents a visualization targeted at a general audience, closely coupled with a statistical NLP algorithm and applied to problems in the real-time communication domain. Interaction is provided through two data change operations (input new data, select an alternative best path). Selecting a new best path is a presentation change as new nodes and edges are highlighted. Presentation may also be changed by enlarging photo nodes to view additional images. View changes are not provided as scaling is automatically calculated to fit the entire lattice on the screen.

A challenging constraint on visualization for real-time communication is that it occur in near real-time. Input is received, processed by the NLP and visualized for the viewer immediately. Visualizations are transitory, so the design should be targeted to facilitate quick interpretation.

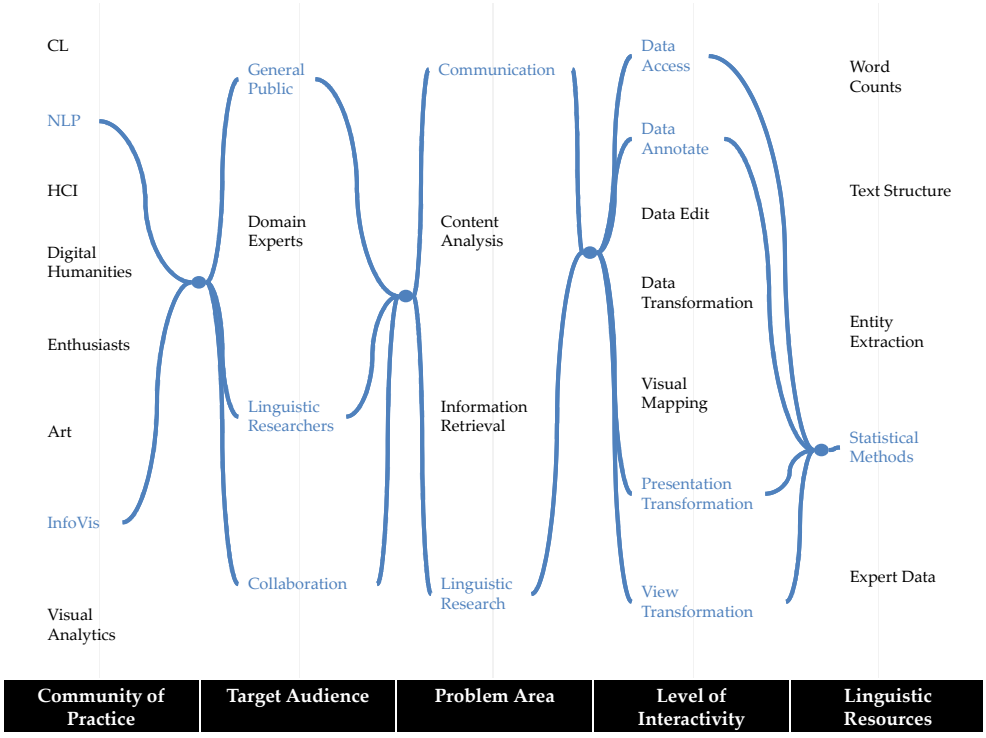


FIGURE 4.12: Uncertainty lattice visualization in the space of linguistic visualization.

Part III

VISUALIZATION FOR CONTENT ANALYSIS

5

VISUALIZING DOCUMENT CONTENT USING WORDNET

It is in the homes of the idlest men that you find the biggest libraries... he knows their titles, their bindings, but nothing else.

— Seneca the Younger (quoted in (Petroski, 2000, p. 26))

One of the diseases of this age is the multiplicity of books; they doth so overcharge the world that it is not able to digest the abundance of idle matter that is every day hatched and brought forth into the world.

— Barnaby Rich (1613)

Visualizations of individual document content is an area of active research and interest as we transition to electronic reading devices and e-libraries. Document visualizations based on simple attributes such as relative word frequency have become increasingly popular tools (Viégas *et al.*, 2009; Viégas *et al.*, 2007). This chapter presents a design study, called *DocuBurst* , which goes beyond previous document content visualizations to grant *spatial rights* to an expert-created structure of language, using it as the backbone for semantically organized visualizations of document content.

Within the *space of linguistic visualization*, the target audience of this design study is a general public or domain expert audience. This design study addresses the problem area of content analysis, specifically by creating semantically-organized views of text. A viewer can interact with DocuBurst in many ways, including accessing (loading) new data, filtering data (altering the data transformation), highlighting and resizing elements (altering the presentation), zooming and panning (altering the view). The linguistic resources used in this study are word counting and the expert-created ontology *WordNet*.

The interface is multi-dimensional, coordinating views of the DocuBurst glyph with a zoomable text selection widget and a full-text panel for detailed reading. Through the use of the ordered visual variable transparency, and the selection of a leafy colour green as the main encoding

hue (Stone, 2003), encoded values within DocuBurst should be intuitively perceived ("walk-up-and-read").

DocuBurst is in some ways the simplest of our realized integrations of linguistics and visualization, using a fixed ontology structure and word counts (normalized by polysemy scores). From a linguistic viewpoint, it illustrates the semantic struction of the language as conceived and refined by expert lexicographers over many years. From an information visualization point of view, it quickly reveals the relative usage of particular nouns and verbs in a long text, and provides for visual summaries at interactively varying levels of detail. It acts as a form of *tag cloud* or *word cloud* with the added advantage of structured, familiar, and additive relations between words. DocuBurst brings surface statistics like word counts together with important linguistic data structures and makes it interactively accessible.

5.1 MOTIVATION

"What is this document about?" is a common question when navigating large document databases. In a physical library, visitors can browse shelves of books related to their interest, casually opening those with relevant titles, thumbing through tables of contents, glancing at some pages, and deciding whether this volume deserves further attention. In a digital library (or catalogue search of a traditional library) visitors gain the ability to coalesce documents which may be located in several areas of a physical library into a single listing of potentially interesting documents. However, the experience is generally quite sterile: the interface presents lists of titles, authors, and perhaps images of book covers. In feature-rich interfaces, page previews and tables of contents may be browsable. If the library contents are e-books, users may even open the entire text, but will have to page through the text slowly, as interfaces are often designed to present a page or two at a time (to dissuade copying). Our motivation in this design study is the possibility to bring some of the exploratory experience to digital libraries by creating interactive summaries of texts which are comparative at a glance, can serve as decision support when selecting texts of interest, and provide entry points to explore specific passages. Providing decision support for selecting a text or region of text is a type of cognitive aid to reduce the burden of *information overload*.

Prompted by the growing information overload problem (see § 2.1) due to the lack of effective technology for managing the ever-increasing

volume of digital text, developing overviews of document content has been an active research area in information visualization for several years. However, reported works such as those discussed in Section 3.4.2 do not make use of existing richly studied linguistic structures, relying instead on simple word counts, alphabetic spatializations (as in tag clouds) or on analytic methods such as latent semantic analysis (DeCamp *et al.*, 2005), which can produce unintuitive word associations. Other document visualizations do not necessarily share our goal of providing thematic content summarization at varying levels of granularity (*semantic zoom*). For example, *TextArc* (Paley, 2002) places all the sentences of a book into a circular layout (see Figure 3.18b). The layout then positions words based on the centroid of all occurrences of that word within the sentences, granting *spatial rights* to the average position of the word in the text. The resulting visualizations provide detail on structure and content without a semantic organization or a consistent view that can be compared across documents.

In DocuBurst, we provide a complement to these works: a visualization of document content based on the linguist-constructed `IS-A` noun and verb hierarchies of *WordNet* (Fellbaum, 1998) which can provide both uniquely- and consistently-shaped glyph representations of documents, designed for intra-document analysis and cross-document comparison.

5.2 ORGANIZING TAG CLOUDS AND MAKING SUMMARIES INTERACTIVE

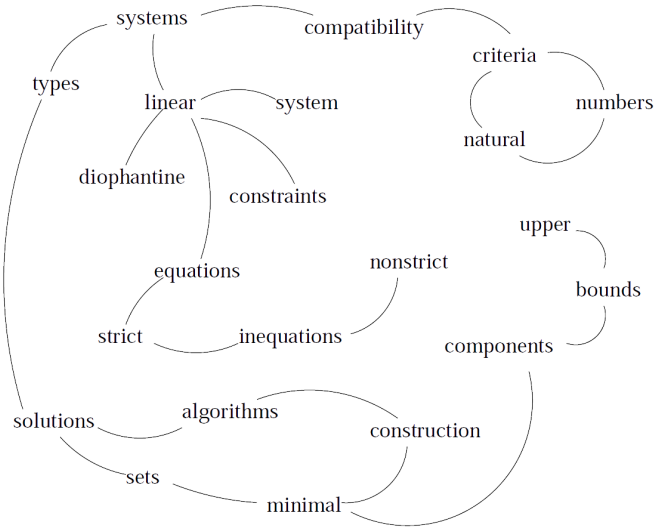
Previous approaches to the problem area of document content analysis span both sides of the *linguistic visualization divide*. From the natural language processing (NLP) side, researchers have developed a variety of methods for extracting keywords and sentences from documents. For example, Mihalcea and Tarau (2004) describe the *Textrank* method, a graph-based ranking model for text processing. *Textrank* is an iterative algorithm: once seeded with initial text units that best define the task (the most important words), relations connecting text units (co-occurrence relations) are identified and traversed to build a graph. While it may seem like the next step could be "then we view that graph", in fact it is "sort vertices based on their final score," essentially abstracting away the graph structure. In the publication about the algorithm, an example sketch of a *Textrank* graph appears (see Figure 5.1a). Yet, the leap is not made to see the potential value of the graph itself as a resource. They do not

investigate methods to automatically draw or interact with that structured representation of the text. For example, the *Textrank* graph itself could potentially be a form of a variable-granularity structured summary.

On the other side of the *divide*, the information visualization (InfoVis) community has invented many document content visualizations, some of which were reported in Chapter 3. In particular, most document content visualizations simply spatialize words alphabetically and visually encode their frequency in the font size (a "word/tag cloud") (Dimov and Mulloy, 2008; Viégas *et al.*, 2009). Hassan-Montero and Herrero-Solana (2006) alter the layout of traditional tag clouds by clustering co-occurring terms into rows (see Figure 5.1b). Through Gestalt proximity perception, the structured version may provide additional cues about word senses (*i.e.*, an ambiguous word like 'bank' occurring in a row with 'river' would be less ambiguous). It could also reveal higher-level themes through the collections of related words. Although the clustered tag cloud therefore has some semantic structure, the structure is statistically determined, not predictable. In related work, we introduced *weighted brushing* as an interactive way to view co-occurrence relations between items in a tag cloud while maintaining predictable alphabetic layout (Dörk *et al.*, 2008). Despite these layout and interaction innovations, Hearst (2008) criticizes tag clouds, writing: "they are clearly problematic from a perceptual cognition point of view. For one thing, there is no visual flow to the layout." She continues, "physical proximity is an important visual cue to indicate meaningful relationships. But in a tag cloud, tags that are semantically similar do not necessarily occur near one another, because the tags are organized in alphabetical order."

DocuBurst represents an attempt to bridge the divide. While we do not use the specific techniques of *Textrank* or co-occurrence tag clustering, DocuBurst does draw on *WordNet*, a structured linguistic resource designed by professional lexicographers. Through *WordNet*, DocuBurst brings meaningful and predictable structure to the spatialization of document content words.

In order to describe the details of DocuBurst, we will now review some specific background relevant to this chapter, followed by an explanation of the visual and interaction design and example scenarios of use.



(a) The *Textrank* algorithm iteratively builds a lexical relationship graph. However, the graph is never visualized — this sketch was published to communicate the method (Mihalcea and Tarau, 2004).

(b) Although the clustered tag cloud has semantic structure, the structure is statistically determined, therefore not predictable for visual search (Hassan-Montero and Herrero-Solana, 2006).

FIGURE 5.1: Contrasting approaches to document content visualization over the linguistic visualization divide. *Reprinted with permission.*

5.3 BACKGROUND ON GRAPH DRAWING

Radial graph-drawing techniques have been previously reported and serve as the basis of this work. Of particular interest are the semi-circular radial space-filling (RSF) hierarchies of *Information Slices* (Andrews and Heidegger, 1998) and the focus + context interaction techniques of the fully circular Starburst visualization (Stasko and Zhang, 2000). The *InterRing* visualization expands on the interaction techniques for RSF trees, supporting brushing and interactive radial distortion (Yang *et al.*, 2002). *TreeJuxtaposer* illustrates methods for interacting with very large trees, where nodes may be assigned very few pixels (Munzner *et al.*, 2003). We adapt techniques such as tracing the path from a node of interest to the root and performing interactive accordion expansion from this work.

5.4 BACKGROUND ON WORDNET

words, synsets, and glosses

Despite the growing dependence on statistical methods, many NLP techniques still rely heavily on human-constructed lexical resources such as WordNet (Fellbaum, 1998). WordNet is a lexical database composed of *words*, *collocations*, *synsets*, *glosses*, and *edges*. *Words* are literally words as in common usage. A *collocation* is a set of words such as 'information visualization' which are frequently collocated and can be considered a unit with a particular definition. For the purposes of this chapter, we will use *words* to refer to both *words* and *collocations* — they are treated equally in the visualization. Sets of synonymous *words* and *collocations* are called *synsets*. *Glosses* are short definitions that the words in a synset share, thus they are definitions of synsets. An edge in WordNet represents a connection between synsets.

Synsets are the most important data unit in WordNet. Throughout this chapter, we will refer to *words* in single quotes (*e. g.*, 'thought'), and synsets using a bracketed set notation (*e. g.*, {thought, idea}). A *word* may be a member of multiple *synsets*, one for each sense of that word. Word senses are ranked, either by order of familiarity (a subjective judgement by the lexicographer) or, in some cases, by using a synset-tagged reference corpus to provide numerical relative frequencies.

Synsets in WordNet are connected by many types of edges, depending on the part of speech (noun, verb, *etc.*). WordNet contains 28 different types of relations, but the most widely used part of WordNet is the

hyponymy (`IS-A`) partial order. An example of hyponymy is {lawyer, attorney} `IS-A` {professional, professional person}. When traversing this graph, we remove any cycles (they are very rare) by taking a depth-first spanning tree at the user-selected root. In this work we focus on the noun hyponymy relationships in English WordNet (v2.1), rooted under the synset {entity} having 73,736 nodes (synsets) and 75,110 edges, and a maximum depth of 14. Verb hyponymy is also supported — that hierarchy is smaller and takes a more shallow, bushier form. In addition, there is no single root verb. The visualizations produced can be generalized to any partial order of a lexicon.

5.4.1 *WordNet Visualization*

Many interfaces for WordNet exist, the most popular of which is the text-based WordNet Search which is part of the publicly available WordNet package. With the exception of the work of Kamps and Marx (2002a,b) and Kamps *et al.* (2004) the existing interfaces for WordNet either provide for drill-down textual or graphical interaction with the data starting at a single synset of interest or provide path-tracing between two synsets (*e.g.*, Alcock, 2004; ThinkMap, 2005). We do not know of any visualization of WordNet that uses the graph structure to enhance a visualization of other data such as document content.

5.5 DESIGN OF THE DOCUBURST VISUALIZATION

The combined structure of WordNet hyponymy and document lexical content is visualized using a radial space-filling tree layout implemented with prefuse (Heer *et al.*, 2005). Traversing the tree from center to periphery follows a semantic path of increasing specificity using the `IS-A` relation. In WordNet, synset members are ordered according to their polysemy count, which WordNet researchers call *familiarity*. Since more familiar words come first, we chose the first word in a synset as the node label. Label fonts are maximized, rotated to fit within the node, and overlap is minimized.

5.5.1 *Linguistic Processing and Scoring*

In order to populate a hyponymy hierarchy with word counts, several pre-processing steps are necessary. Starting with raw text, we subdivide the text into *tiles* based on the pre-existing structure, such as section headings. If no structure is detectable, we break the text into roughly coherent topic segments using a segmenter (Choi, 2000). For each tile, we label parts of speech (`NOUN`, `VERB`, etc.) (Brill, 1993). Nouns and verbs are then extracted and stemmed (*e. g.*, books $\rightarrow$ book, going $\rightarrow$ go) using a morphological processor (Didion, 2003). Punctuation is omitted. If short word sequences, noted in WordNet, are found in the document, the words are combined into a collocation, and treated as a single word.

Next we look up in which WordNet synsets the (*word, part-of-speech*) pairs occur. Because pairs usually occur in multiple synsets, we do not perform word sense disambiguation. Instead, we divide the word count amongst the available synsets. If WordNet supplies relative sense frequency information for a word, we use this to distribute the count. Otherwise, we distribute the count weighted linearly by sense rank. This results in weighted occurrence counts that are not integers, but the overall results more accurately reflect document content. By dividing the counts, we dilute the contribution of highly ambiguous terms. The full text of tiles and their associated (*word, part-of-speech, count*) triples are then read into the data structure of the visualization.

5.5.2 *Visual Encoding*

Node Size

Within the radial tree, angular width can be proportional to the number of leaves in the subtree rooted at that node (*leaf count*) or proportional to the sum of word counts for synsets in the subtree rooted at that node (*occurrence count*). The leaf count view is dependent on WordNet and so is consistent across documents. The word count view maximizes screen space for synsets whose words actually occur in the document of interest, thus the shape, as well as node colouring, will differ across documents. Depth in the hyponymy tree determines on which concentric ring a node appears. The width of each annulus is maximized to allow for all visible graph elements to fit within the display space.

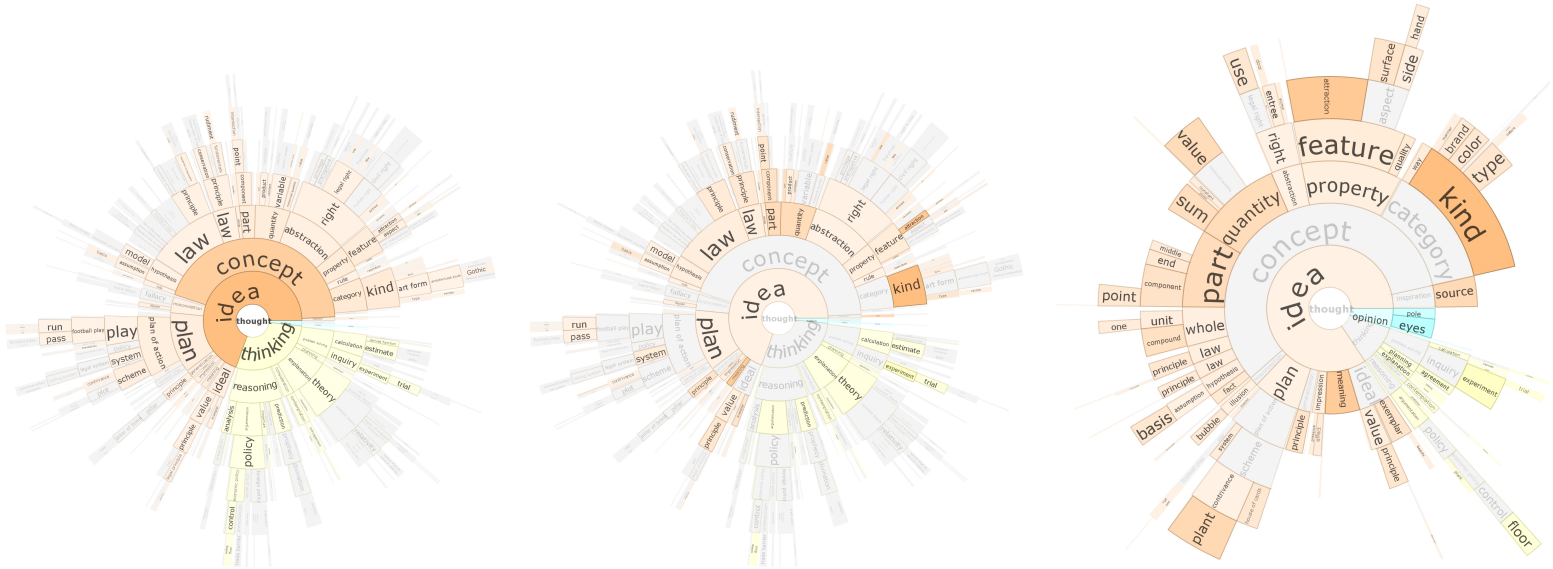
Node Colour

It is possible to look at multiple senses of a word in one view. Views rooted at a single word contain a uniquely coloured subtree for each synset (sense) containing that word. In contrast, trees rooted at a single synset use a single hue. Since luminance variation in the green region of the spectrum is the most readily perceived, it is the first colour choice (Stone, 2003, p. 30). Gray is used for nodes with zero occurrence counts, since their presence provides a visual reminder of what words are not used.

Transparency is used to visualize relative word or synset count. Similar to the concept of value, transparency provides a range of light to dark colour gradations, thus offering ordered (Bertin, 1983) and "pre-attentive" (Ware, 2004) visuals (Table 2.3.3). Highly opaque nodes have many occurrences; almost transparent nodes have few occurrences. Word senses that are more prominent in the document stand out against the more transparent context.

Two ways to visualize word occurrence are provided: single-node and cumulative. In the *single-node* visualization, only synset nodes whose word members occur in the document are coloured. In the *cumulative* view, counts are propagated up to the root of the tree. In both views, transparency is normalized so maximum counts achieve full opacity. When multiple documents are visualized, the cross-document maximum is used to set the scale. These modes support a gradual refinement of focus. The cumulative, or subtree, view uses the association of words into synsets and synsets into a hyponymy tree to aggregate counts for related concepts. Similar to the TreeJuxtaposer techniques for visualizing differences embedded deep in a large tree (Munzner *et al.*, 2003), by highlighting the entire subtree containing the node, salient small nodes can be more easily located, even if hidden from view by a filter. The single-node view reveals precise concepts in the document and supports the selection of synsets whose word members appear in the document being analyzed. In addition, for a fully expanded graph, the single node view may highlight nodes that are otherwise too small to notice. The subtree and cumulative views are compared in Figure 5.2.

While transparency is an effective visual method for distinguishing large differences and trends, it is difficult to read exact data values using it. To facilitate the exact reading of synset occurrence counts for the selected text tiles, we provide a dynamic legend (see Figure 5.3). Brushing

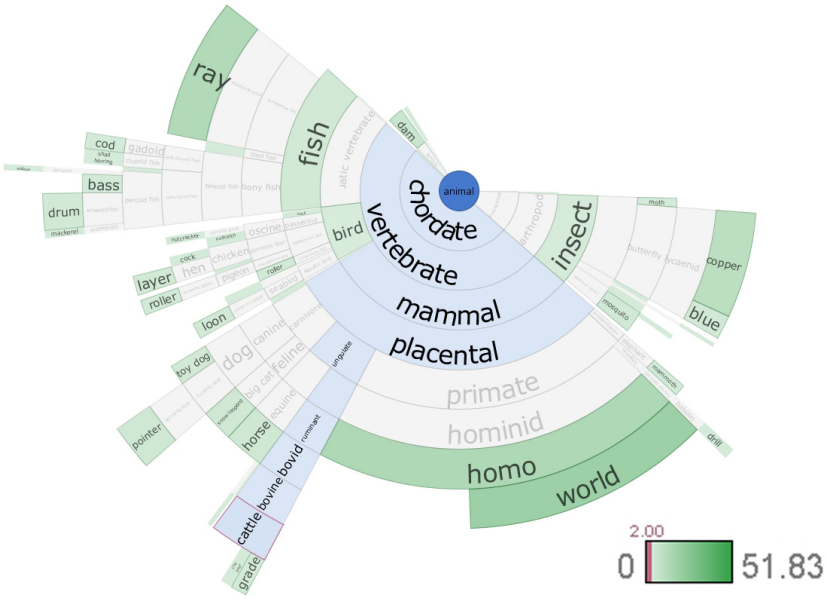


(a) Node fill transparency proportional to number of occurrences of words in subtree (cumulative view). Node size based on leaf count.

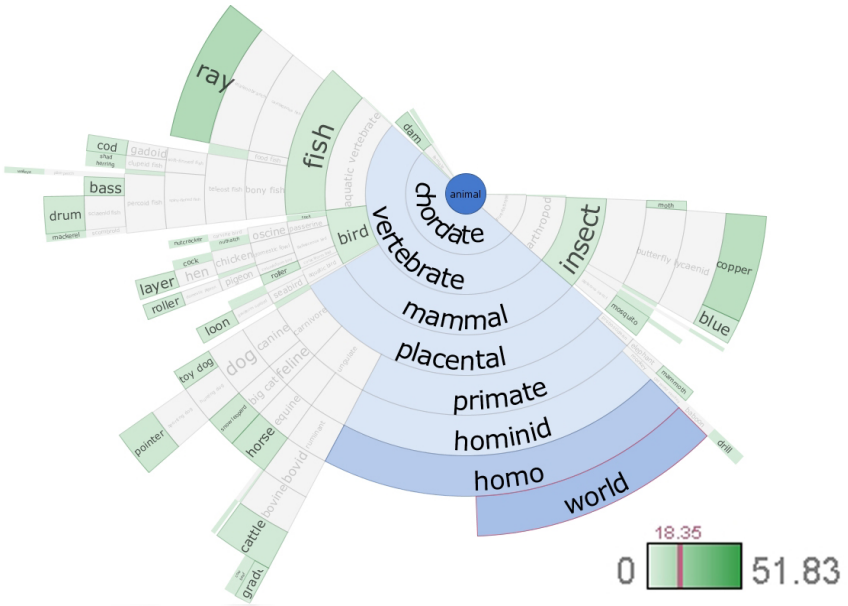
(b) Node fill transparency proportional to number of occurrences of words in a single synset (single node view). Node size based on leaf count.

(c) Single node view, node size proportional to occurrence count total for subtree rooted at that node.

FIGURE 5.2: Colour and size encoding options for DocuBurst. Shown here is a science textbook rooted at 'thought'; node hue distinguishes the synsets containing 'thought'.



(a) Synset {cattle, cows, kine, oxen} highlighted.



(b) Synset {world, human race, humanity, mankind, man} highlighted.

FIGURE 5.3: DocuBurst of a general science textbook rooted at {animal}. Single-node colouring and occurrence count sizing were used with zero-occurrence synsets hidden. The mouse hover point is revealed by blue trace-to-root colouring. The dynamic legend (bottom right of each image) is enlarged to show the detail.

over a node reveals its score and where that score falls within the range of the data.

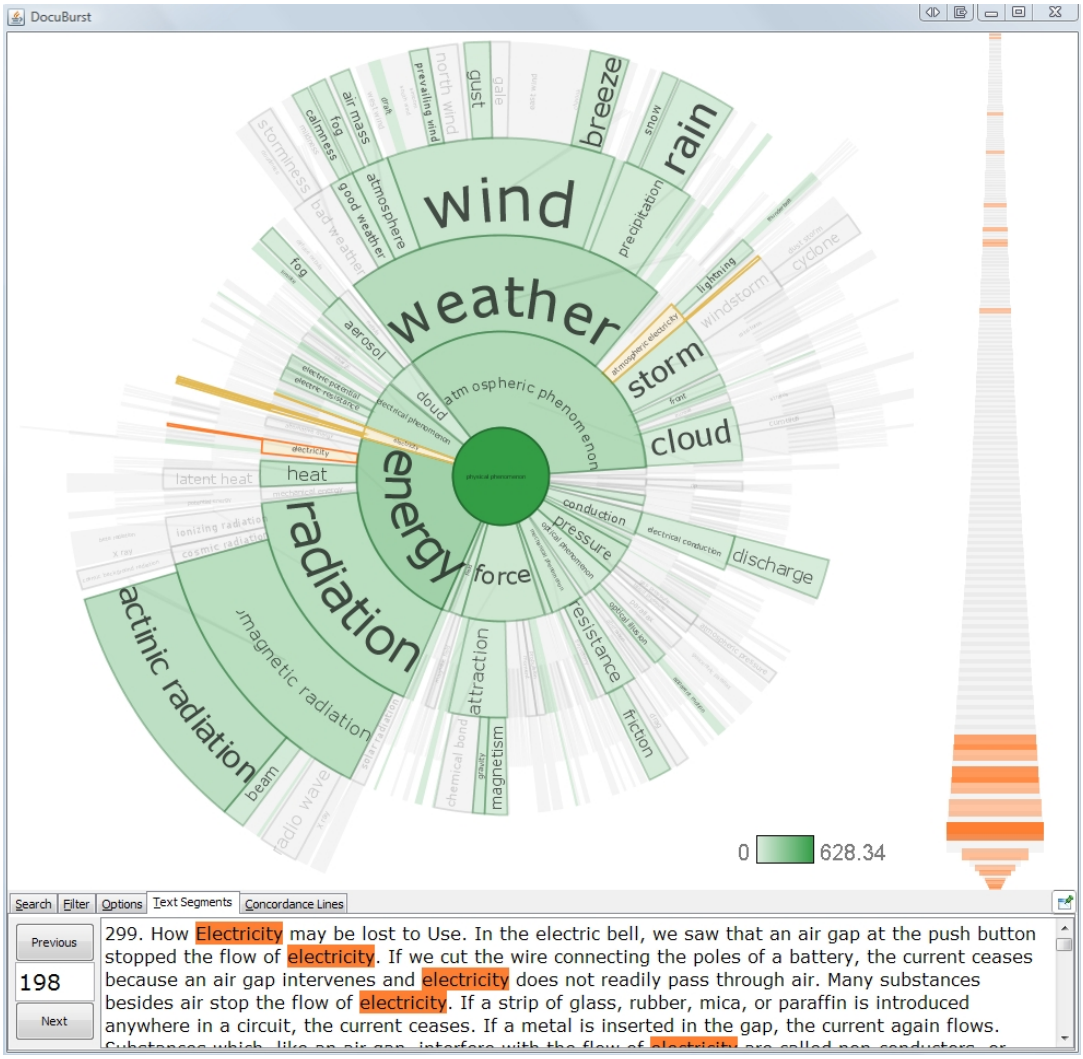
5.5.3 *Interaction*

A root node can be a word, in which case its immediate children are the synsets containing that word. Alternatively the visualization can be rooted at a synset. Root nodes in the visualization are selectable by searching for either a word or synset of interest. Once a root is chosen, the visualization is populated with all its hyponyms.

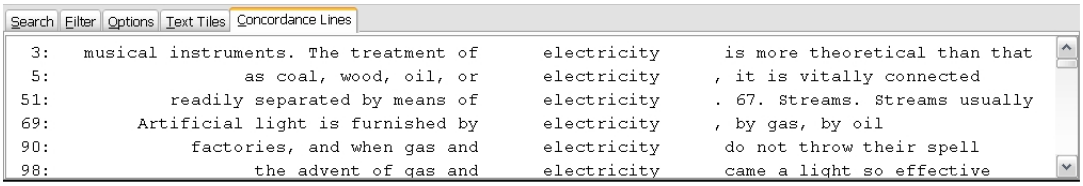
As there are more than 70,000 English noun synsets in WordNet, techniques to abstract and filter the data are important. First, we provide a highlight search function which visually highlights nodes whose label matches any of the given search terms. *Highlight nodes* have a gold background and border, and a darker font colour, drawing attention to even the smallest of search results. The transparency of the highlight (gold) background is attenuated to the word occurrence counts so as to not disrupt this data-carrying value and to provide for stronger pop-out of search results with high occurrence counts.

Second, we implement a generalized fisheye view (Furnas, 1986) that collapses all subtrees which are more than a user-specified distance from the central root node. Changing this distance-based filter allows for a semantic zoom, creating visual summaries of varying specificity. The presence of non-zero word occurrence counts within collapsed subtrees is indicated by using the cumulative colouring, in which counts are propagated to the root. Optionally, all highlight nodes can be exempted from the distance filter (by increasing their *a priori* importance in the degree-of-interest (DOI) function), effectively abstracting the graph to all synsets within a given distance from the root or highlight nodes (see Figure 5.4).

Double clicking on a node of interest restricts the visualization to the hyponyms of the node's synset; double right-clicking reverses this action by reloading the graph at the parent of the clicked node, thus providing bi-directional data navigation through the hyponymy relation. To create more space for the details of the children of a given synset, the angular width of a node and its subtree can be manually increased using the mouse wheel. This increase provides a radial detail-in-context view which causes the node's siblings to be correspondingly compressed. Changes to



(a) Linked visualizations for details-on-demand.



(b) The details window showing the concordance lines for the selected synset.

FIGURE 5.4: A search for 'electricity' reveals synsets containing that word (gold hue). On selecting a node, the distribution of the word is revealed in the tile browser. Selecting a tile reveals the text with occurrences of the selected synset highlighted.

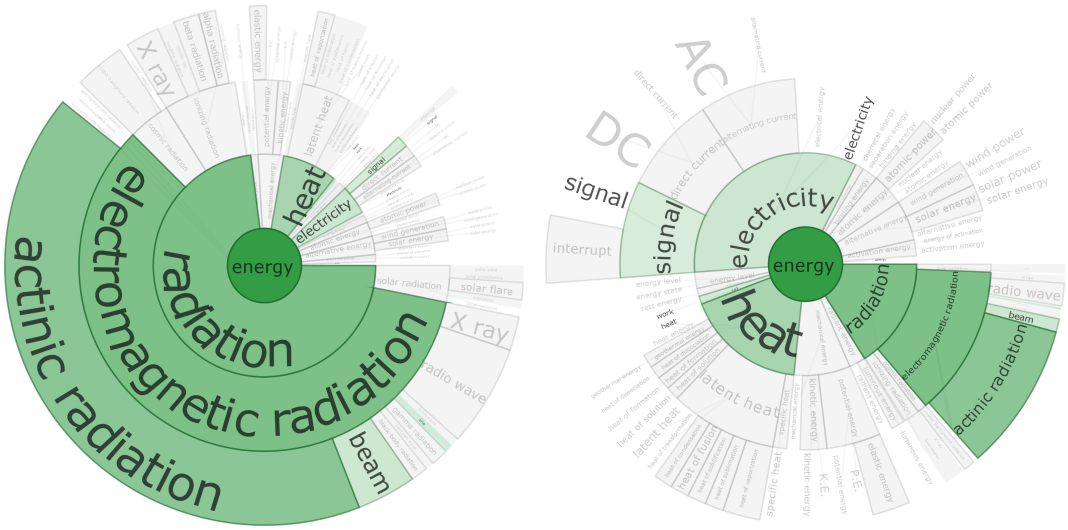


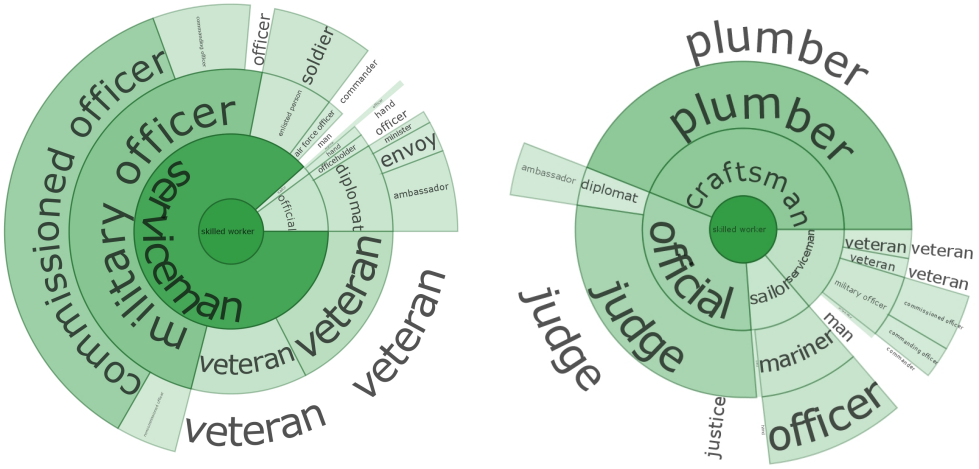
FIGURE 5.5: DocuBurst of a general science textbook rooted at {energy}. At right, mouse wheel interaction was used to shrink the angular width of the subtree rooted at {radiation} and expand the subtree under {electricity} exposing the previously illegible node {signal}.

a node's angular width affect its children equally and its siblings in an inverse manner (see Figure 5.5).

The visualization can be based on selected subsections of the document. The initial view is based on all text tiles in the document, but a selection can limit the tiles from which counts are drawn. Unrestricted visual pan and geometric zoom of the display space are also supported, as well as a zoom-to-fit control to reset the pan and zoom to a best-fit for the currently visible tree. Rendering is dependent on the zoom factor: node borders are not rendered when the nodes are very small, and labels are not rendered when they would not be legible. All highlighting, navigation, and emphasis interactions are provided in real time.

5.5.4 *Accessing the Source Text*

The full text can be read through accessing the text tiles at the bottom of the interface. To navigate the text, we use a linked visualization: the text tile browser. Rectangles representing the text tiles appears in a linear, vertical array to the right of the DocuBurst. A fisheye distortion (Bederson, 2000) facilitates navigation and selection of text tiles within this list. Clicking any tile brings it into view. Furthermore, this visualization



(a) First 2008 debate (b) Third 2008 debate

FIGURE 5.6: DocuBurst glyphs rooted at {skilled worker} reveal that the traditional U.S. focus on military officers and veterans was eclipsed in the third U.S. Presidential debate by discussions of plumbers.

can be used to see occurrence patterns in the document. By clicking nodes in the DocuBurst visualization, synsets and entire subtrees can be selected. Text tiles in which selected synsets appear show as varying intensity of orange in the text tile browser, depending on number of occurrences in the tile. Occurrences of those synsets and words are also highlighted in the full text window.

Displaying concondence lines in a keyword-in-context (KWIC) is a standard linguistic analysis tool in which all occurrences of a word of interest are extracted and displayed with their left and right N (usually 5) context words (Luhn, 1960). KWIC lines for selections are extracted and shown in the concordance window. Patterns of orange in the tile browser can indicate how localized concepts are in the document. For example, in Figure 5.4, we see that {electricity} appears more frequently toward the end of the document. We can use the tile browser and full text window to quickly find occurrences of the terms of interest in context. By clicking the text tile rectangles in the tile browser, we find, in the tile detail window, that there is a chapter on 'electricity' at the end of the book.

5.6 EXAMPLE: DOCUMENT COMPARISON

DocuBurst can be used to compare multiple documents. Trees rooted at the same synset but coloured based on different texts will reveal relative frequency differences between them. While the other examples in this chapter were visualization of a high-school general science text book, we also can apply the technique to other forms of electronic text. In Figure 5.6 we applied DocuBurst to the transcripts of two 2008 U.S. presidential debates. Note that to ensure comparability when viewing multiple documents, colour scaling is based on the maximum count for visible nodes across all documents.

A high-level view of the debates rooted at {person} revealed strong colour for the {leader} and {serviceman, military personnel, military man} subtrees. Drilling down revealed {senator} is a descendant of {leader} (both participants were senators). Attention to military issues and veterans is also expected given current conflicts. Examining the third debate showed an additional region of colour under the {craftsman} subtree. Further investigation, by switching to the occurrence count size function, revealed a dramatic shift in concentration within the {skilled worker} subtree. Military people became relatively less important compared to craftspeople — specifically, plumbers. This is the effect of Senator McCain's focus on "Joe, the Plumber" in the third debate, and was the genesis point of this phrase which dominated the remainder of the campaign.

5.7 SUMMARY

We have already briefly reviewed linguistic visualization for document content analysis in Section 3.4.2. Recall that visualizations of document content take two common forms: synoptic visualizations for quick overviews and visualizations specialized for discovering patterns within and between documents. Specialization in the type of document used as input further divides the reported research: books and long documents, historical documents, multilingual texts, and computer-mediated communication archives such as emails, instant messages, and threaded discussions. In this space, DocuBurst focuses on long texts, such as books, and provides a visualization that is simultaneously synoptic, comparative, and allows for deeper intra-document analysis of occurrence patterns.

Feature	Starstruck	Compus	Word Tree	TextArc	Arc Diagram	Gist Icons	Tag Clouds	TileBars	FeatureLens	DocuBurst
Semantic	N	N	N	N	N	N	N	N	N	Y
Cluster	N	N	N	N	N	Y	N	N	N	Y
Features	P	Y	N	N	N	P	P	Y	P	P
Suggest	N	N	N	Y	Y	Y	Y	N	Y	P
Overview	N	Y	N	Y	Y	N	Y	Y	N	Y
Phrases	N	Y	Y	N	Y	P	Y*	P	Y	N
Read	P	N	P	Y	P	P	P	P	Y	Y
Freq	Y*	N	N	Y	N	Y*	Y	Y*	Y*	Y*
Compare	Y	Y	P	N	N	Y	Y	Y	Y	Y
Search	N	N	Y	Y	P	Y	Y	Y	Y	Y
All words	Y	N	Y	Y	Y	Y	Y	Y	Y	N
Pattern	P+	Y	Y*	Y	Y	P+	P+	Y*	Y*	Y+

TABLE 5.1: Comparison of features available (Y), possible with a trivial extension (P), or not possible (N) in document visualizations. * denotes cases that only visualize a selected subset of words; + denotes a coordinated visualization. Rows and columns ordered by $|N + 1/2P|$.

To compare document visualizations, we order a list of the types of features document visualizations have provided from most unique to DocuBurst to the most common in other visualizations:

SEMANTIC indicate word meaning

CLUSTER generalize by clustering words into concepts

FEATURES reveal extracted features (*e. g.*, emotion)

SUGGEST suggest interesting focus words/phrases

OVERVIEW provide quick overviews of an entire text

PHRASES can show multi-word phrases

ZOOM support varying the semantic or graphical detail

READ drill-down to original text

FREQ reveal frequency of individual words

COMPARE compare multiple documents

SEARCH search for specific words/phrases

ALL WORDS can show all parts of speech

PATTERN reveal patterns within or between texts

Table 5.1 summarizes how these features relate to DocuBurst and other well known text visualizations. Notice that only DocuBurst provides some

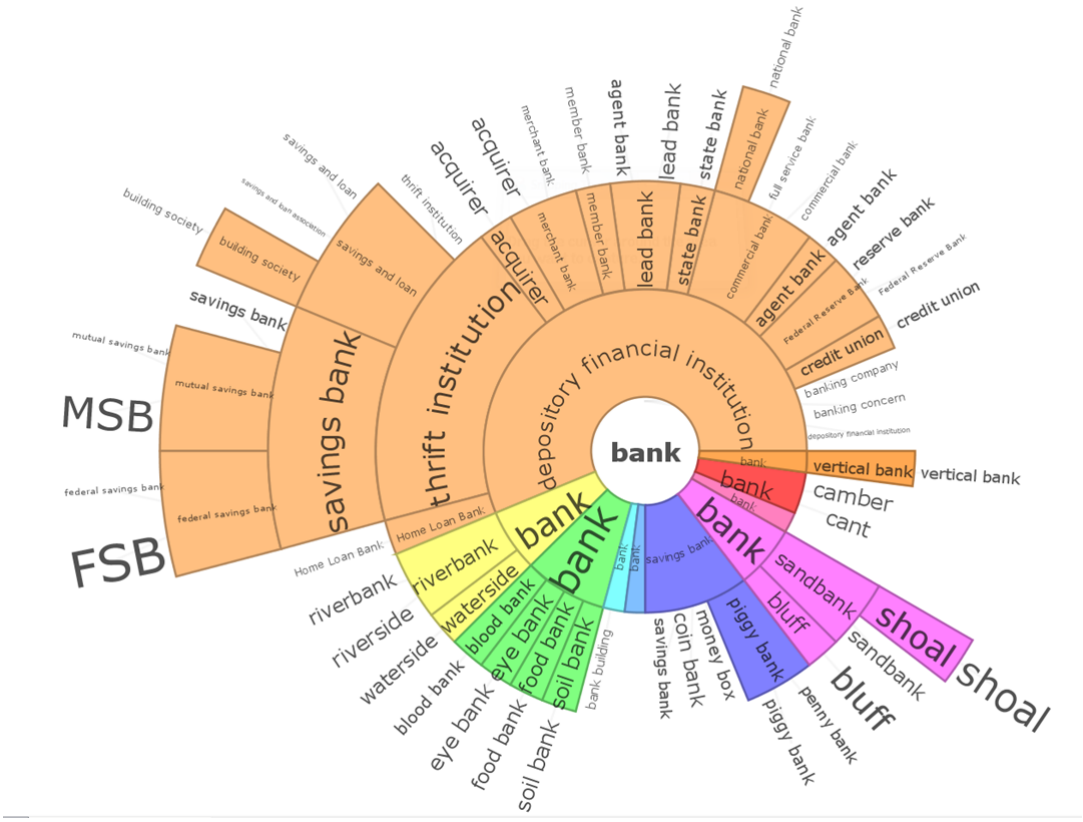


FIGURE 5.7: DocuBurst, without word count information, can be used as a form of visual dictionary. This figure shows the various senses of the word 'bank'.

reflection of semantics through integrated word definitions and the use of a semantically-defined linguistic structure. Only DocuBurst and *Gist Icons* (see §3.2.1) provide word clustering into higher concepts; however in Gist Icons the groups are only one level deep and based on statistical measures whose meaning may not be readily apparent to a reader. Note that all visualizations which provide overviews of entire texts suffer from screen real-estate issues with large texts.

Many organizations and individuals have expressed interest in DocuBurst. Usage requests include as a way to manage document repositories (*e.g.*, NATO LEGAL DEPARTMENT, the University of Calgary Library, the Meandre Project, Boeing Research, Verilogue, and several other information technology firms). Potential use as a document comparison tool for forensic linguistics was also suggested by developers at the RCMP. DocuBurst has been featured in the media, including the Toronto Star (Bigge, 2007), CBC Radio (Watt, 2008), and Fairchild Television (Liu *et al.*, 2008).

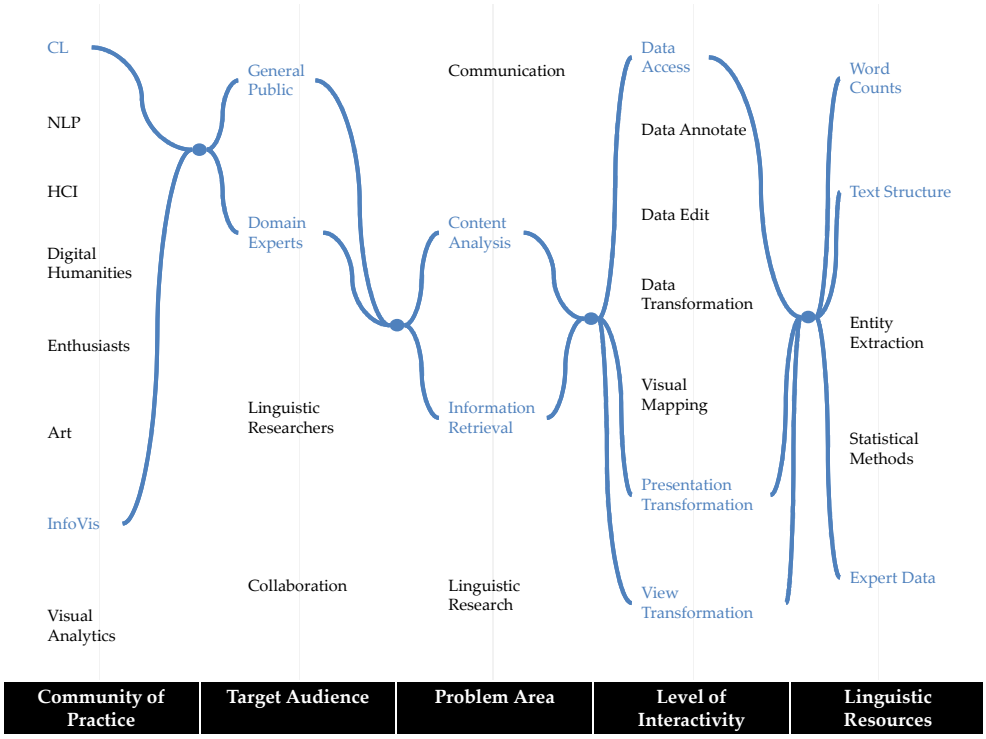


FIGURE 5.8: DocuBurst in the space of linguistic visualization.

Beyond use as a document content visualization, DocuBurst has attracted the attention of educators interested in using the RSF glyphs as a visual dictionary for teaching purposes — for example, a teacher wrote to us: "DocuBurst has invaluable potential for writing and vocabulary development at the secondary level." An example of a DocuBurst glyph used as a way to explore the various definitions of 'bank' appears in Figure 5.7.

DocuBurst provides an overview visualization of document content which is based on a human-centered view of language whereas previous works were based on linguistically simpler, derivative statistical analyses. While the statistical analyses previously used to generate semantically arranged document content visualizations are lingusitically simpler than our expert-design resource, they are arguabely also more complex to use. An average reader may not see clearly why a particular statistical algorithm associates seemingly quite different terms, making extracting

the higher-level concept challenging. DocuBurst may suffer from word sense ambiguity problems, but the structure is predictable, regular, and interpretation of the relations is straightforward. The visual design is grounded in established research on human abilities in colour perception. Semantic and geometric zooming, filtering, search, and details-on-demand provide a visual document summary, revealing what subset of language is covered by a document, and how those terms are distributed. The position of DocuBurst within the space of linguistic visualization is summarized in Figure 5.8.

6

PARALLEL TAG CLOUDS FOR FACETED CORPORA

Jargon serves lawyers as a bond of union: it serves them, at every word, to remind them of that common interest, by which they are made friends to one another, enemies to the rest of mankind.

— Jeremy Bentham (Bowring, 1843, p. 292)

While DocuBurst (Chapter 5) is a content analysis visualization of text for a general audience, linguistic visualizations can also be tailored to specific audiences, while remaining highly coupled to the data processing. In this chapter we describe Parallel Tag Clouds, a lexical visualization created for legal academics (*domain experts*) to aid in their investigation of differences between the types of cases heard in the various U.S. Circuit Courts of Appeal. This visualization is closely coupled to several linguistic manipulations, from simplistic techniques like stemming (and reverse-stemming) to more the more computationally intensive G^2 statistic calculated over all words. The resulting visual interface retains easy access to the underlying text.

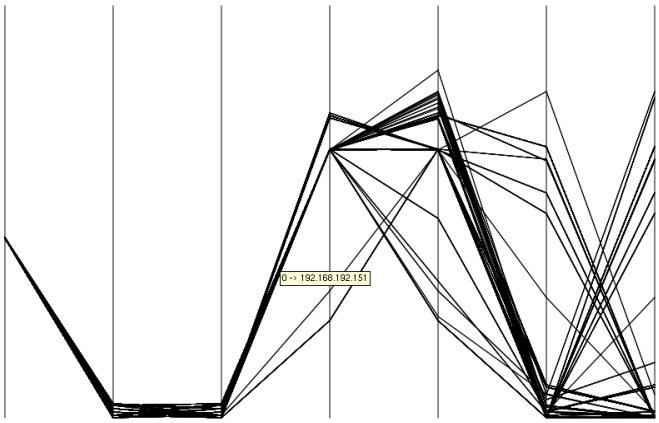
6.1 MOTIVATION

Academics spend entire careers deeply analyzing important texts, such as classical literature, poetry, and political documents. The study of the language of the law takes a similar *deep reading* approach (Tiersma, 1999). Deep knowledge of a domain helps experts understand how one author's word choice and grammatical constructs differ from another, or how the themes in texts vary. While we may never replace such careful expert analysis of texts, and we likely will never want to, there are statistical tools that can provide overviews and insights into large text corpora in relatively little time. This sort of *distant reading* on a large scale, advocated by Moretti (2005), is the focus of this work. Statistical tools alone are not sufficient for distant reading analysis: methods to aid in the analysis and exploration of the results of automated text processing are needed, and visualization is one approach that may help.

Of particular interest are corpora that are faceted — scholars often try to understand how the contents differ across the facets. Facets can be understood as orthogonal, non-exclusive categories that describe multiple aspects of information sources. For example, how does the language of Shakespeare's comedies compare to his tragedies? With rich data for faceted subdivision, we could also explore the same data by length of the text, year of first performance, *etc.* Documents often contain rich metadata that can be used to define facets, such as publication date, author name, or topic classification. Text features useful for faceted navigation can also be automatically inferred during text pre-processing, such as geographic locations extracted from the text (Dörk *et al.*, 2008), or the emotional leaning of the content (Gregory *et al.*, 2006).

In the legal domain, a question often asked is whether different court districts tend to hear different sorts of cases. This question is of particular interest to legal scholars investigating *forum shopping* (the tendency to bring a case in a district considered to have a higher likelihood to rule favourably), and this was the initial motivation for this investigation. Our research question, then, is whether we can discover distinguishing differences in the cases heard by different courts. We address this question through examination of the written decisions of judges. The decisions of U.S. Courts are officially in the public domain, but only recently have high-quality machine-readable bulk downloads been made freely available (Malamud, 2008). Providing tools to augment our understanding of the history and regional variance of legal decision making is an important societal goal as well as an interesting research challenge. Beyond our specific case study in legal data, we are interested in broader issues such as easing the barriers to overview and analysis of large text corpora by non-experts, and providing quick access to interesting documents within text collections.

Our solution combines text mining to discover the *distinguishing* terms for a facet, and a new visualization technique to display and interact with the results. Parallel Tag Clouds bridges the *linguistic visualization divide* by blending the interactive visual techniques of parallel coordinate plots (Inselberg and Dimsdale, 1990) (see Figure 6.1a) and tag clouds (recall *Wordle*, Figure 3.14) with statistical models of lexical significance created by the natural language processing (NLP) community (see Figure 6.1b). Rich interaction and a coordinated document browsing visualization allow Parallel Tag Clouds to become an entry point into deeper analysis. In the remainder of this chapter we will describe Parallel Tag Clouds in



(a) An example of a Parallel Coordinates Plot (Inselberg and Dimsdale, 1990) — each axis represents a data dimension, each line represents a data item. *Image in the public domain.*

Rank	Unigram	(-2λ)	Bigram	(-2λ)	Trigram	(-2λ)
1	Slovenia	319.48	federal army	21.27	Slovenia central bank	5.80
2	Yugoslavia	159.55	Slovenia Croatia	19.33	minister foreign affairs	5.80
3	Slovene	87.27	Milan Kucan	17.40	unallocated federal debt	5.80
4	Croatia	79.48	European Community	13.53	Drnovsek prime minister	3.86
5	Slovenian	67.82	foreign exchange	13.53	European Community countries	3.86

(b) A list of significant, or distinguishing terms for a document given a reference corpus (Lin and Hovy, 2002). *Reprinted with permission.*

FIGURE 6.1: Contrasting approaches to corpus overview over the linguistic visualization divide.

comparison to existing methods of corpus visualization, the interaction and coordinated views provided to support analytics, our text mining and data parsing approach, and some example scenarios of discovery within the legal corpus.

6.2 BACKGROUND

In order to describe the details of Parallel Tag Clouds, we will now review some specific background relevant only to this chapter, followed by an explanation of the visual and interaction design and example scenarios of use.

6.2.1 *Exploring Text Corpora*

For the purposes of our work, we define facets in a corpus as data dimensions along which a data set can be subdivided. Facets have a name, such as 'year of publication' and data values such as `1999` which can be used to divide data items. Attention to faceted information has generally been focused on designing search interfaces to support navigation and filtering within large databases (*e. g.*, Healey, 2007). In faceted browsing and navigation, such as the familiar interfaces of *Amazon.com* and *Ebay.com*, information seekers can divide data along a facet, select a value to isolate a data subset, then further divide along another facet. For our purposes, we divide a document collection along a selected facet, and visualize how the aggregate contents of the documents in each subset differ.

While there are many interfaces for visualizing individual documents and overviews of entire text corpora (*e. g.*, Collins *et al.*, 2009; Havre *et al.*, 2002; Wattenberg and Viégas, 2008; Wise *et al.*, 1995), there are relatively few attempts to provide overviews to differentiate among facets within a corpus. One notable exception is the radial, space-filling visualization of Rembold and Späth (2006) for comparing essays in a collection (§3.2.5). Others have created variants on tag clouds to contrast two specific documents (*e. g.*, Clark, 2008a; IBM Research, 2009). None of these comparative visualizations focus on both visualization and appropriate text mining as a holistic analytic system, but rather use simple word counts to illustrate differences among documents. The work most related to Parallel Tag Clouds is *Themail* (Viégas *et al.*, 2006), a system for extracting significant words from email conversations using statistical measures and visualizing them using parallel columns of words along a timeline (see Figure 3.4.2). The visualization approach of Parallel Tag Clouds shares the focus on discovering differentiating words within subsets of a corpus, and visualizes text along parallel columns of words. However, Parallel Tag Clouds can reveal significant *absence*, or underuse of a word, as well as significant *presence*, or overuse. We augment the *Themail* approach with connections between related data subsets. Parallel Tag Clouds are also visually similar to the connected lists view of Jigsaw (Stasko *et al.*, 2007) (§3.2.9), however Parallel Tag Clouds use size-weighting of words in the display.

Shneiderman and Aris (2006) have previously explored the contents of faceted legal document databases using matrix-based visualizations to reveal the number and type of data items matching each facet value. Our

work differs in that we seek to aggregate and visualize the contents of the data items, not only their presence or absence. A matrix visualization approach would not be appropriate as our word-selection method, described later, seeks to maximize the differences between corpus subsets. Rather than the single vertical column of words that a words × facets matrix would contain, our approach allows the entire space to be filled with a wide variety of words.

VisGets, or visualization widgets, have been used to explore faceted collections of Web-based streaming data (Dörk *et al.*, 2008) (§3.4.3). Facets are filtered using scented visual widgets (Willett *et al.*, 2007) appropriate for the data type, providing both an overview of the available data items and a method to drill down along several facets simultaneously. A tag cloud VisGet consists of a traditional tag cloud summarizing all available documents — text differentiation along a facet is only achieved through interactive brushing. The goal of VisGets is to provide coordinated overview and navigation tools in a faceted information space, where our work is customized to providing meaningful differentiating overviews across facets within large amounts of textual data.

Finally, the *Authorlines* visualization (Viégas *et al.*, 2004) provides an overview of individual messages using arrays of circles, sized according to message length. We borrow this visual encoding and extend it to small multiples of bar charts in the document browser coordinated view, linked to the Parallel Tag Cloud.

6.2.2 *U.S. Circuit Court Decisions*

The words of the iconoclast Bentham (see Chapter epigraph) were not the last written on the topic of legal language. Law and language meet in many academic ways: forensic linguists help solve crimes, judges make semantic rulings on unclear contract wording, and social scholars take a high-level view, studying the language of lawyers and judges (Tiersma, 1999). By analyzing the written decisions of the U.S. Circuit Courts of Appeal, we hope to shed light on thematic and potentially linguistic differences between subsets of the data. Differences in word usage between courts has been previously studied using legal databases as a source for historical lexicography (Shapiro, 2003). However, in that work, text-based searches provided information on particular words of interest. Through

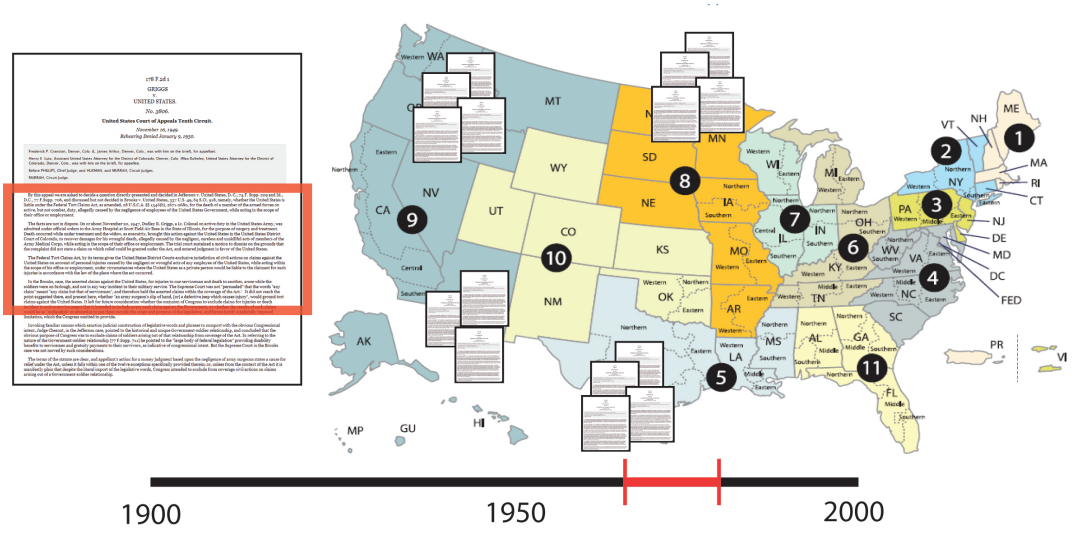


FIGURE 6.2: Structure of U.S. Circuit Court data: Each court case contains various sections (left). U.S. Court *Circuits* are multi-state regions (right). Court cases are also time-stamped, allowing filtering by time (bottom).

text mining and visualization, we select words of interest and provide a broad overview as an entry point to deeper analysis.

The U.S. Circuit Courts of Appeal are made up of 12 regionally-based court divisions (numbered First through Eleventh, plus the DC Circuit) and the Federal Circuit, which hears cases of national relevance, such as patent-related appeals (see Figure 6.2). This data contains of 628,000 court decisions, each labeled by circuit. The judgments are faceted, because they can be organized along several dimensions, such as the lead authoring judge, the decision length, the date of the decision, or whether the lower court was upheld or overturned. For our purposes, we parse the raw data and divide it into subsets by circuit, but we could equally well subdivide along other facets.

Each court decision is made up of several parts: the court name, the parties involved in the case, the date of the hearing, the date of the decision, the authoring and concurring judges, the main decision, optional concurring and dissenting opinions, and optional footnotes (see Figure 6.3). In the data we obtained, most sections were pre-labeled in XML, but there were many errors, such as the date coded as the court name. The breaks between main opinion and consenting/dissenting opinions were not labeled. We cleaned the data by re-parsing the full text and labeling each section using regular-expression matching. Our visualization supports

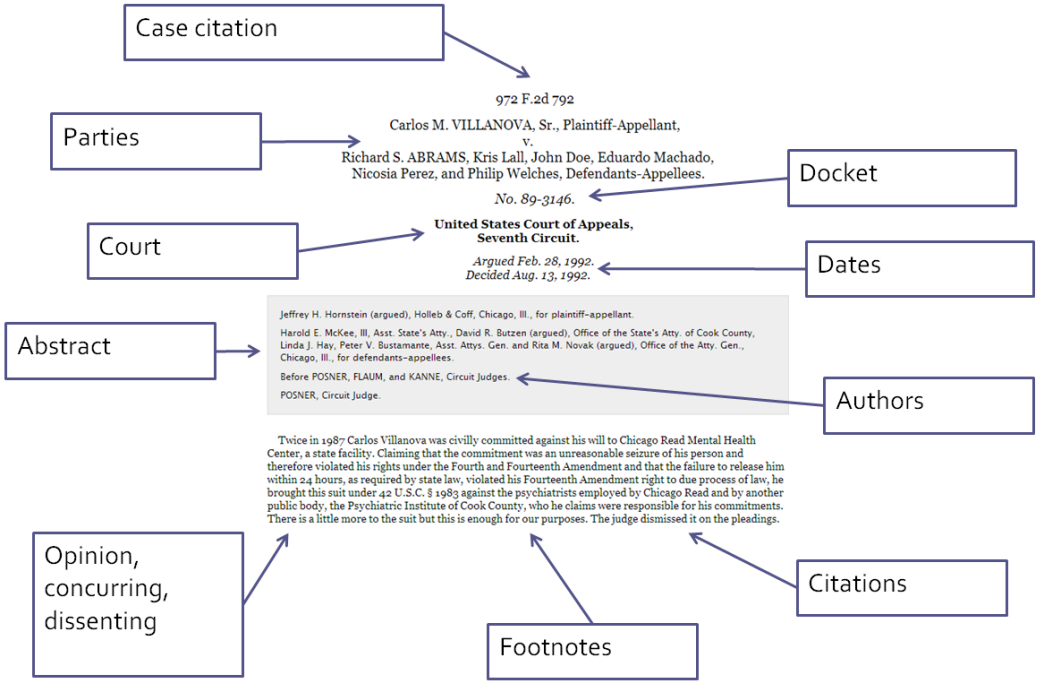


FIGURE 6.3: Each document, or *case* contains several mandatory elements, such as the court name, date of decision, and parties involved, *etc.* Other sections, such as dissenting opinions, may or may not be present.

viewing Parallel Tag Clouds by dividing the data along the `court` facet and loading cases from a selected time period. Comparisons can be made between different courts across any of the textual parts of case data: the entire case, party names, main opinions, concurring opinions, dissenting opinions, and footnotes.

In the following description and accompanying illustrations we will discuss examples pertaining to discovering *distinguishing words* within the written decisions of each circuit court. While the method for discovering words of interest is discussed in detail in Section 6.4, it is sufficient for the following explanation to think of the selected words as characteristic or representative of the court in which they appear, when compared to the remainder of the corpus.

6.3 PARALLEL TAG CLOUDS

Parallel Tag Clouds combine layout techniques from parallel coordinates with word-sizing techniques from tag clouds to provide a visualization for comparing large amounts of text. The basis of the visualization is the arrangement of words of interest into parallel columns, one for each distinct subset of the data across a facet of interest (see Figure 6.4). Of several visual encodings tested by Bateman *et al.* (2008), font size was the best way to convey importance, so we use it to encode a pre-assigned score. We scale by font size rather than the area of a word, as area gives undue visual prominence to short words. Words that are common across columns are connected by nearest-neighbor edges. Edges are drawn with varying width, sized relative to the words at the endpoints to reinforce relative size differences across columns. An important distinction between Parallel Tag Clouds and parallel coordinate plots is that in Parallel Tag Clouds edges may bypass a column (parallel coordinates axis) when a word is not present.

Through informal trials, we have found that the edges provide useful information about the degree of overlap among columns in general, but also tend to increase the complexity of the visualization and reduce the legibility of the words. To reduce the problem, all edges are drawn as 'stubs' that connect to each endpoint word and fade to transparency between the words. These edge stubs indicate the presence and direction of a connection, while not further cluttering the display and disrupting legibility. An exploration of the many alternative designs for easing edge

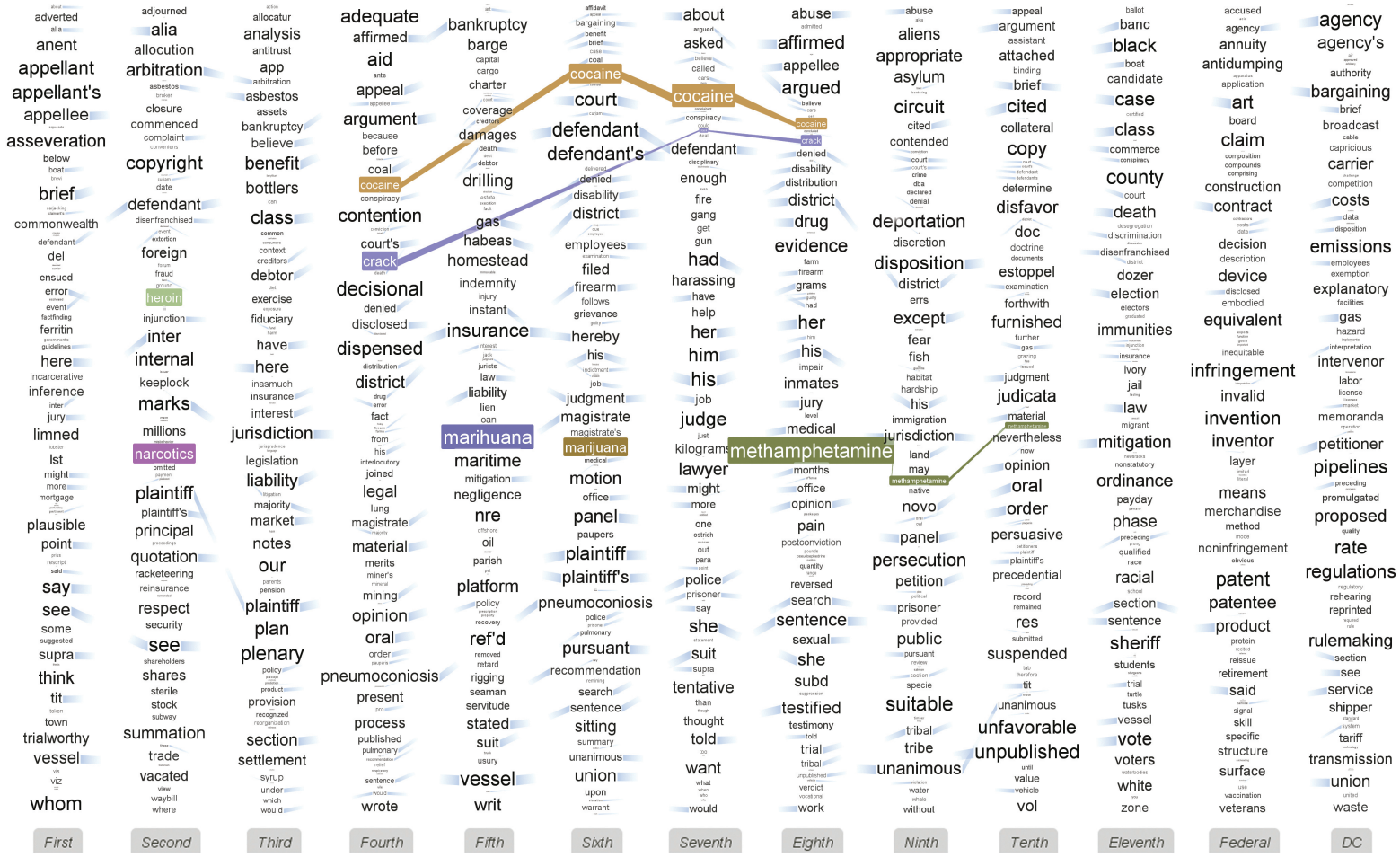


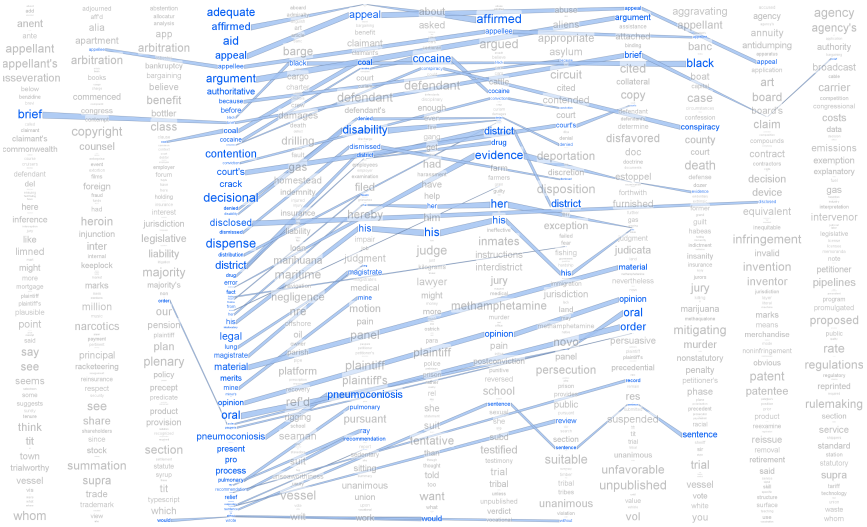
FIGURE 6.4: A Parallel Tag Cloud revealing the differences in drug prevalence amongst the circuits.

congestion while maintaining some visual overview of connections can be found in Appendix C. We clarify edges through interaction: when the pointer hovers over or selects a term, all occurrences of that term, and the connecting edges between them, are fully drawn and highlighted. Additionally, an entire column may be selected, making edges attached to this column visible and revealing all terms it shares with other columns (see Figure 6.5). This provides the ability to *drill across* the corpus: by finding a term of interest in one column, one can easily discover others in which it is present and expand exploration laterally.

We experimented with two arrangements of words: alphabetical and ordered by size. Alphabetical arrangement was preferable for several reasons. While ordering by size offers the ability to identify the most significant words in each column (by reading across the tops), the layout is not space-efficient. Inter-column spacing must be wider as all the largest words cluster at the top. Alphabetical ordering distributes words of different sizes throughout the vertical space, allowing columns to be closer together. That is, we can place columns so close that two words at the largest size will overlap, because two words at the largest size are unlikely to be adjacent. Additionally, alphabetical ordering supports visual scanning to determine the presence of words of interest.

6.3.1 *Sizing by Rank and Score*

In a dense visualization which has the special requirement that nodes are words which must be legible, maximal use of space is crucial. In order to maximize space usage, the default view of Parallel Tag Clouds sizes words by their *rank* in the column. The result is that each column, assuming it has the same number of words, will have the same distribution of sizes, thus the same length. This provides for efficient usage of the vertical space and maximizes the average font size for all columns. However, information is lost about the relative magnitude of the underlying scores which generate the ranks. As we use a significance threshold to select words for the visualization, every word in the view is significant, so arguably maximizing size over precision may be preferable. Sizing proportional the maximum score across all columns may result in some columns becoming very small if their relative scores are small. This can be informative, and we provide this view as an option. For example, using our scoring function of how distinguishing a word is for a particular court,



(a)

(b)

FIGURE 6.5: Brushing on the label for a column highlights all edges connected to that column, and where they lead. In (a) we can see the Fourth has connections to many of the circuits to the West, particularly the Sixth, but lacks similarities to the Northeastern circuits First through Third. In (b) we see very few of the terms significant to the Federal Circuit are also significant to other circuits.

FIGURE 6.6: Sizing by score reveals that the Federal Circuit (second from right) is the most different from other courts, and that the word 'patent' is overall the most differentiating word in the selected time period of the corpus.

a shorter column would represent a court which is overall less different from the entire corpus than a court with a long column (see Figure 6.6).

Outliers can be distracting to an analyst, and detrimental to the ability to visually to discriminate among other data items which are closer in value. Analysts often desire the ability to interactively remove distracting data items from view (Wattenberg, 2006). Right-clicking a word in the Parallel Tag Cloud removes it from the view, causing remaining items to rescale to fill the space.

6.3.2 *Exploring Documents in the Corpus*

The initial population of the Parallel Tag Cloud occurs by selecting a facet of interest to subdivide the corpus. In our implementation this is fixed: we subdivide by court. However, we only visualize courts of interest. For example, the Federal Circuit is quite different from the others, as it hears mainly patent-related cases, so it may be omitted. Data is also filtered by selecting a time period. We use scented widgets (Willett *et al.*, 2007) to simultaneously allow for courts of interest to be selected while encoding how many cases are in that court for the selected time-frame. After selecting a data range, the tag cloud is populated with the top N words in each column, where N is pre-set to be maximal to allow for readable font sizes given the current window height, but may be adjusted. Larger values of N will introduce the need to scroll the visualization to explore the entire tag cloud.

Text overview visualizations are generally most useful if an analyst can interactively obtain additional information about specific words. As our implementation of Parallel Tag Clouds draws on a large collection of documents, we provide the ability to select terms of interest and explore their distribution throughout the document collection. When a term of interest is selected, a coordinated document browser visualization is populated with bar charts representing the individual documents in which that term occurs, organized in rows by a second facet in the data, such as by year. The height of the bar is proportional to the number of occurrences of the term in that document. When multiple terms are selected, each is assigned a unique highlight colour on the tag cloud, and the document glyphs become stacked bar charts. Multiple selections are treated as an `AND` query, preventing an overload of document results. Results are grouped

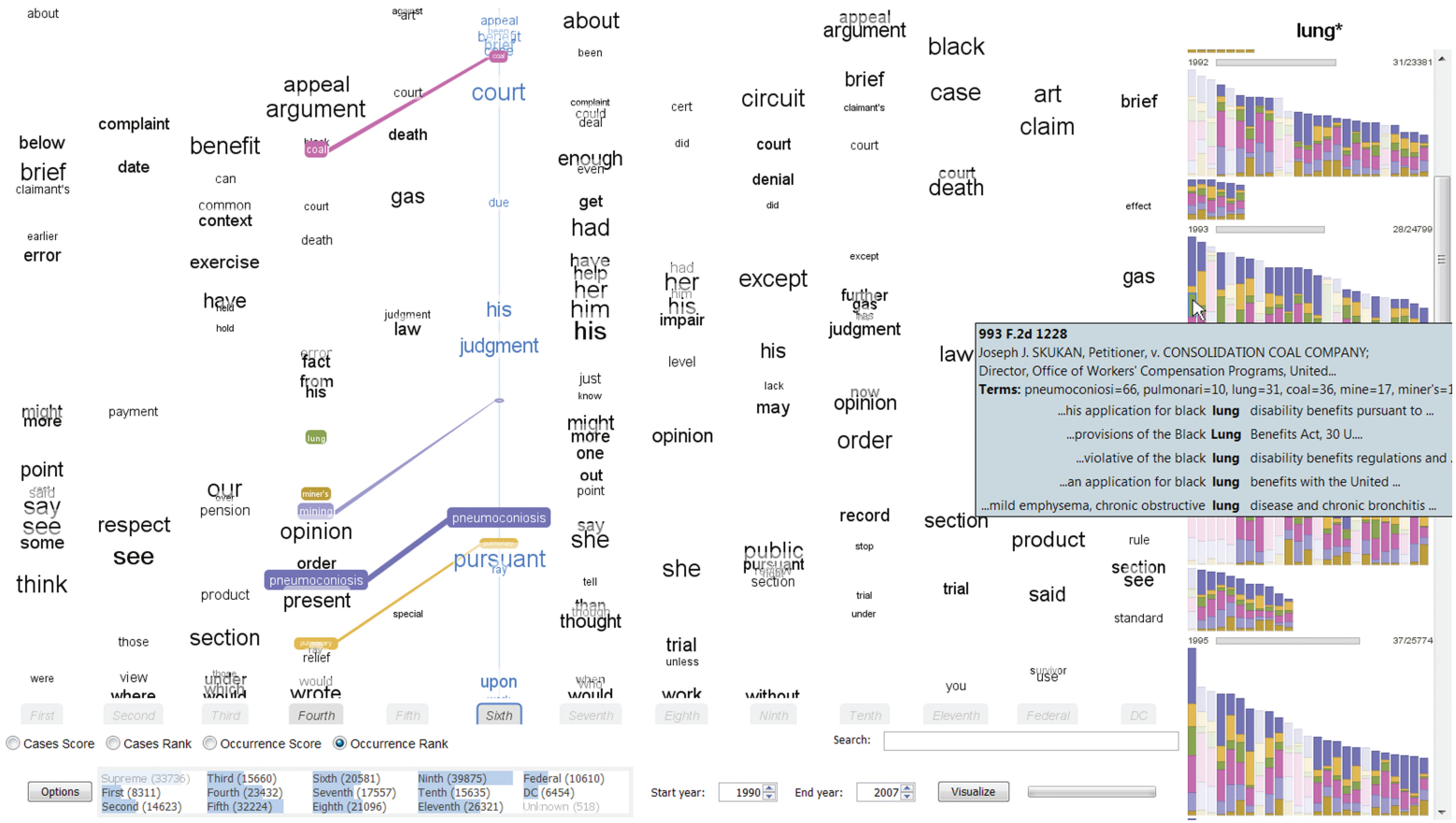


FIGURE 6.7: Selected terms in the Parallel Tag Cloud reveal litigation related to the coal mining-related disease pneumoconiosis in both the Fourth and Sixth Circuits. The document browser at right reveals the distribution of selected terms in individual cases over time. The Fourth and Sixth Circuits are selected in the Parallel Tag Cloud, causing documents from circuits other than these to become semi-transparent in the document browser. The mouse pointer brushes over a section of a stacked document bar. The words in that document are enlarged on the Parallel Tag Cloud and all other words are removed. A tooltip shows examples of the hovered word used in context.

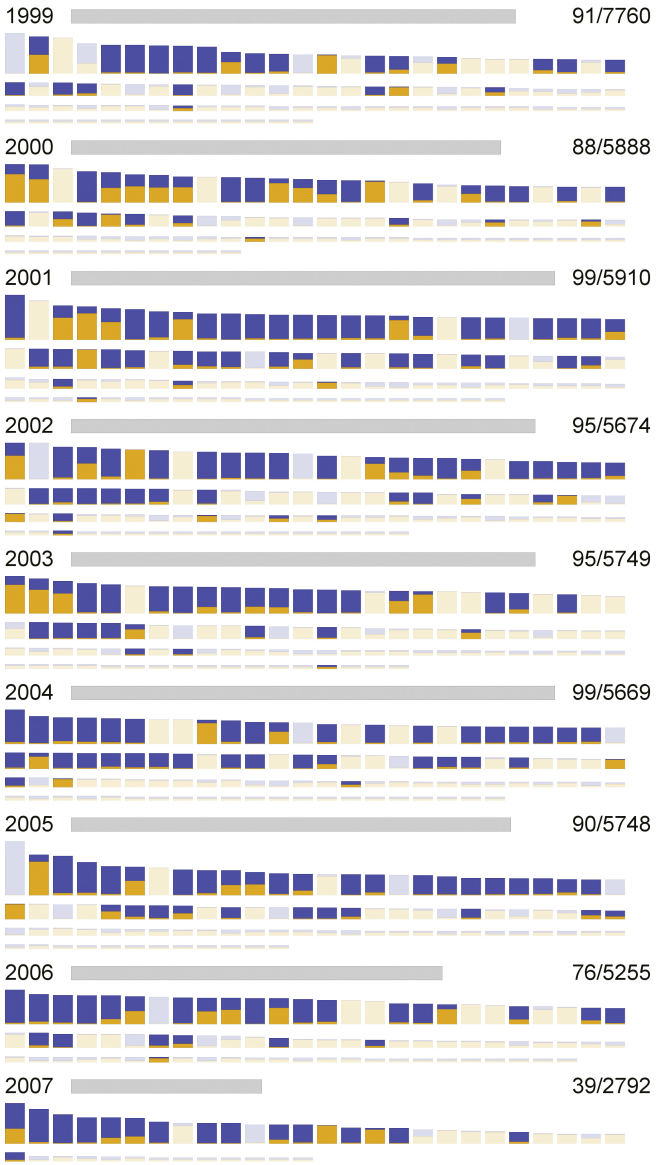


FIGURE 6.8: Detailed view of the document browser for 'patent' (blue) and 'claim' (gold). These terms often co-occur. We have selected the Federal Circuit, so cases not in the Federal Circuit are faded out.

by year and ordered largest to smallest. A maximum of 100 results per year are shown.

To provide a complete picture of the results, horizontal 'distribution bars' beside the year labels show the relative number of documents matching the search terms and what portion of these are hidden. The total distribution bar length is broken into two portions: the documents which are visualized as glyphs in the document browser contribute to a fully opaque grey block, hidden documents contribute to the semi-transparent region. Thus the distribution bar reveals the proportion of available documents which match the search query, and over which years, even when the individual document glyphs are hidden to save screen space.

Views are interactively linked: brushing across a document icon in the document browser highlights all the terms occurring in that document which are also in the Parallel Tag Cloud (see Figure 6.7). Words are highlighted by increasing the font size and fading out words that are not in the document. Additionally, we highlight which corpus subset contains the document by drawing that column in blue. This interaction provides a lightweight form of document content overview, although only words which are already in the Parallel Tag Cloud are shown. Tooltips in the document browser reveal detailed case data, including the citation, parties, authoring judge, and a keyword-in-context (KWIC) table showing examples of the selected word in use (Luhn, 1960).

We provide filtering of items in the document browser by selecting columns of interest in the Parallel Tag Cloud. When any column is selected, documents from non-selected columns become partially transparent (see Figure 6.7, right). We retain the presence of faded document glyphs to give an indication of what proportion of total documents containing the selected terms come from the selected corpus subsets.

Finally, an analyst may wish to read a particular document in detail. Double-clicking a document glyph opens the source document in a Web browser. Additionally, the full text of the document is visualized in a separate tab using a *Many Eyes* tag cloud (Viégas *et al.*, 2007).

In Figure 6.8 we show a detailed view of the document browser for 'patent' and 'claim', with the Federal Circuit selected. The distribution of glyphs reveals that while the range of search word occurrences per case (bar height) is quite large, the majority of the Federal Circuit cases contain many occurrences of these words. Further investigation through hovering on some of the short bars revealed that they are not cases of patent

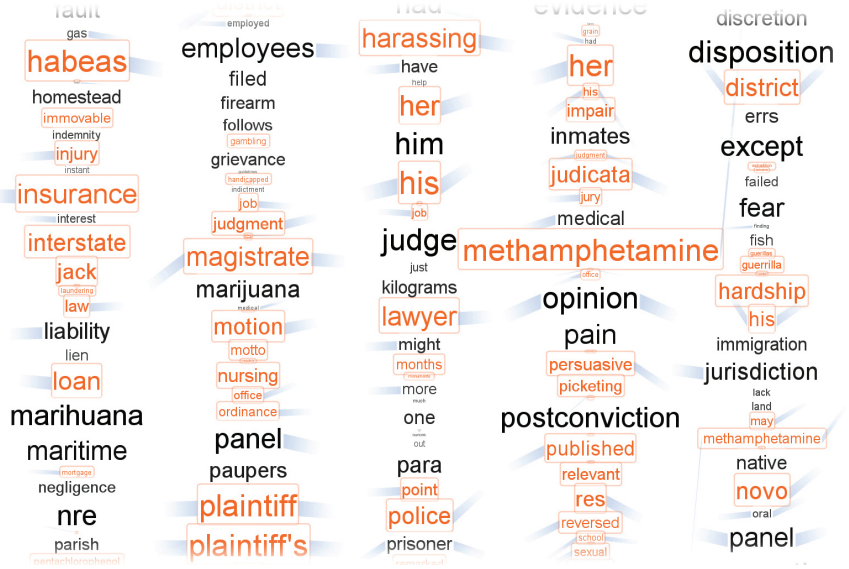


FIGURE 6.9: Data changes are highlighted in orange. Here we see the emergence of 'methamphetamine' (second column from right) as we move from 1990–1995 to 1995–2000. 'Marijuana' is present in both time periods.

litigation, but rather uses of the word 'patently'. The most significant cases in 1999 and 2003 are not from the Federal Circuit, which is curious and invites further reading as to why a patent case was heard outside the Federal Circuit.

6.3.3 *Revealing Change*

As we provide interactive ways to filter the data backing the visualization, such as selecting a time range, we also provide for visual highlighting of changes in the visualization when new data is loaded. New words can appear, for example, by selecting a different time period to extract from a large corpus, or by adjusting the method by which words are selected. When the data filters are adjusted, some words may be removed from view, while others are added. We visually highlight all deleted words and animate them out of view by increasing their size while simultaneously fading them to transparent. This provides a hint at what has been removed. In a second stage of animation, we reveal words that have been added. These remain highlighted until the analyst cancels the highlights through clicking an icon on the interface (see Figure 6.9).

6.4 MINING FACETED CORPORA

The most common approach to visualizing text as tag clouds is to count the word frequencies (*e. g.*, Feinberg, 2008). While this provides a relatively meaningful overview of a single text, or even a collection of texts treated as one, word frequency does not have sufficient distinguishing power to expose how subsets of a text collection differ. While one could compare multiple frequency-based tag clouds for subsets of a document collection, it is likely that these tag clouds will highlight similar words. If there is enough text in each subset, on the order of millions of words, each frequency-based tag cloud will start to approximate the distribution of terms in the domain generally. That is, the most common words will be similar in all data subsets. We would be unlikely, for example, to find much to distinguish among different court districts, where the legal language common among them will dominate. Such an approach may be appropriate for comparing text collections where dramatic differences in common terms were expected, or when similarities are desired.

The information retrieval community has long been interested in discovering words that make a document or collection of documents distinct from the background noise of a large corpus. These distinguishing terms are often given higher weight as index terms for a document, for example. Distinguishing terms have other uses, such as comparing corpora for similarity and homogeneity (Kilgariff and Rose, 1998), or subdividing text automatically based on changes in distinguishing terms (Hearst and Karadi, 1997). While there have been many uses for discovering distinguishing terms in a corpus in applications such as information retrieval and automatic summarization, interactive analysis tools for investigating distinguishing terms in a corpus have not been reported. In fact, Rayson and Garside (2000) explicitly call for analyst intervention, claiming that simply identifying terms is not enough: human expertise is needed to understand why terms may be identified and if the reason is truly meaningful in the analysis context. They suggest 'the researcher should investigate occurrences of the significant terms using standard corpus techniques such as KWIC'. Interactive visualization, such as Parallel Tag Clouds, can offer more powerful analytic avenues for deeper investigation over standard corpus techniques.

There are a multitude of measures reported in the NLP community for scoring and ranking distinguishing terms, and indeed much argument

about their relative quality (*e.g.*, Dunning, 1993; Kilgariff and Rose, 1998; Morris *et al.*, 2004; Rayson and Garside, 2000). Measures such as TF-IDF (term frequency-inverse document frequency) (Spärck Jones, 1972) are commonly used to select distinguishing terms for a paragraph, document, or collection of documents. The *Themail* visualization (Viégas *et al.*, 2006) (Figure 3.4.2) uses a variant of TF-IDF to collect distinguishing terms from a corpus of emails. While TF-IDF is an appropriate measure for detecting distinguishing words in a text sample against a reference corpus, it cannot highlight *significant absence*, nor do the scores it returns reflect a measure of significance for which there are reliable thresholds. A common word that does not appear in a document has a TF-IDF score of zero, the same as a rare word that does not appear.

Often, multiple metrics are applied in weighted combination or in sequence by re-ranking term lists. While multi-statistic methods may return improved results, the numerical scores are difficult to interpret. Indeed, the common practice is to heuristically choose a threshold and discard everything below it (Inkpen and Hirst, 2002). We choose to follow Rayson and Garside (2000) and use a G^2 statistic, which is able to approximate χ^2 for words occurring 5 times or more. The G^2 metric can be interpreted as a measure of significance: higher G^2 corresponds a smaller p value. Or, to simplify: G^2 tells us the probability that the frequency of occurrence of a word in one corpus differs significantly from another.

For low frequency words, Dunning (1993) shows that p values obtained using a G^2 statistic to lookup from a χ^2 tables can be off by several orders of magnitude. However, Morris *et al.* (2004) suggests a method to approximate p-values for low frequency events using the linear relationship between the negative of the natural logarithm of p-values computed from Fisher's exact test and log likelihood ratio scores.

Some have argued that applying the statistic in hypothesis testing is not appropriate given the non-random nature of text: some significant differences among texts is always expected, making the null hypothesis non-interesting (*e.g.*, Kilgariff and Rose, 1998; Rayson and Garside, 2000) . While this is certainly true for any two random documents, our texts are subsets of a larger corpus in the same domain, and each subset of text we compare consists of millions of words. With the increased sample size, the expectation that the subsets will converge on the same domain-specific overall word distribution grows. Thus, differences found may

be significant. While hypothesis testing may be theoretically arguable for judging significance of G^2 scores, we follow Mueller (2008) and use a $p < 0.01$ threshold of significance when visualizing distinguishing terms. This allows us to reduce the number of identified terms, as we cannot visualize all words, and to provide useful hints to an analyst comparing the relevance of terms identified by our statistical tests. The G^2 statistic is calculated using the following contingency table and equations:

	Target Subset	*Remainder of Corpus*	*Total*
$C(word)$	a	b	$a+b$
$C(other\ words)$	$c-a$	$d-b$	$c+d-a-b$
Total	c	d	$c+d$

$$E_1 = c*(a+b)/(c+d) \tag{6.1}$$

$$E_2 = d*(a+b)/(c+d) \tag{6.2}$$

$$G^2 = 2*(a*ln(a/E_1)+b*ln(b/E_2)) \tag{6.3}$$

where $C(word)$ is the count of the target word, and E_1 and E_2 are the expectation values for the word frequency in the target subset and the remainder of the corpus respectively. To find a significance level of $p < 0.01$, we use Moore's conservative approach, without assuming the > 5 word occurrences needed for reliable approximation by χ^2 tables:

$$G^2 \approx -2*ln(p)+2.30 \tag{6.4}$$

which gives us a G^2 threshold of 11.15. We employ a *Sidak correction* for repeated testing to adjust the significance levels. We assume $50,000$ repeated trials (the approximate number of word forms compared on a typical run of our system) and adjust p as follows:

$$p' = 1-(1-p)^{1/k} \tag{6.5}$$

where p' is the adjusted level of significance, and k is the number of trials. This gives us an adjusted p' of $2.01*10^{-7}$, which has a corresponding G^2

cutoff of 33.13, which we use as the threshold in our significance testing. If $a < E_1$, we know the statistic represents a lower than expected frequency of occurrence, otherwise the actual occurrence is higher than expected.

While our prototype of Parallel Tag Clouds uses the G^2 statistic, our visualization is neutral to the scoring method applied to the terms: the visual techniques would work equally well for a frequency-based metric as for the frequency-profiling techniques we have described.

6.4.1 *Occurrence and Case-Based Scoring*

Experiences with *Themail* (Viégas *et al.*, 2006) revealed that techniques for identifying distinguishing words are prone to identifying words which are highly occurring in a particularly long document, but may not be distributed throughout the corpus subset under investigation. For example, in our analysis, 'voters' was identified as a distinguishing term for the Fifth Circuit, however, further investigation revealed a single very lengthy decision on an election-related class action which used the word 'voters' extensively. While this may be of interest to an analyst, it is important to support easy discovery of terms which have high occurrence but low distribution within the corpus subset. To address this, we measure two G^2 scores for each word: an occurrence-based score, and a case-based score. In the case-based measure, we populate the G^2 contingency table by counting how many individual documents (court cases) the word appears in at least once. As we will demonstrate in the analysis, the case-based measure identifies terms which occur in a larger than expected *number of cases* in a corpus subset, rather than an absolute number of occurrences. Both measures have analytical significance and reveal complementary information about a corpus. We provide for viewing Parallel Tag Clouds based on either measure but we also provide for interactive tools to allow for the two forms of score to be compared for a particular word of interest. Additionally, the document browser can quickly reveal the distribution of a selected word within the corpus.

6.4.2 *Data-Rich Tooltips*

While our visualization can only reveal a limited number of words per parallel column, our word scoring measures assign values for all words for all corpus subsets. For example, our measure of the distinguishing

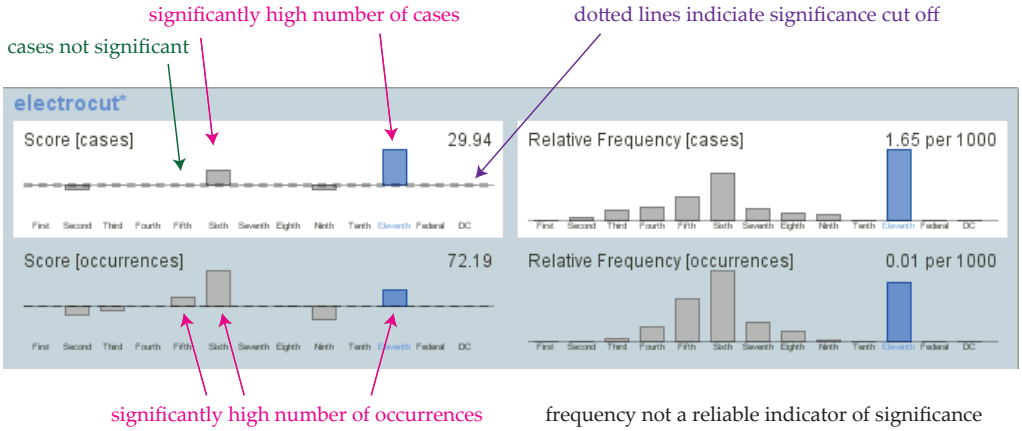


FIGURE 6.10: The bar chart tooltip provides word score details.

nature of a term can identify words which occur more often than expected, or less often than expected. Due to space considerations, we choose to only show words which occur more often than expected. We also calculate occurrence and case-based measures, but can size the tag cloud based on only one. We provide for data-rich graphical tooltips which use bar charts to reveal the score and the normalized frequency of occurrence for a term across all subsets of the corpus, for both occurrence- and case-based measures. The column in which the word under the mouse appears is highlighted in blue to provide easy reference. Threshold lines reveal the G^2 significance threshold, and bars below the threshold are faded out. These tooltip graphs can quickly reveal where a word which is distinguishing in a particular corpus subset is unusually unpopular in another, and whether a term identified using occurrence-based scoring also appears in a significantly high number of cases in the selected court.

In Figure 6.10, we show a tooltip created by hovering on the word 'electrocuted' in the Eleventh Circuit. We can see that this term has a significantly high score for the Fifth, Sixth, and Eleventh Circuits when based on the occurrence count, and occurs less than expected in the Second, Third, and Ninth (bottom left). Note that the significance bars are at the baseline due to the large scale, so are not visible. However, based on the case scores (top left), only the Sixth and Eleventh Circuits have a significantly high score. This indicates that the occurrence-based score in the Fifth Circuit must be due to a few cases with a high number of mentions of this term.

6.4.3 *Data Filtering*

Parallel Tag Clouds, as with any word-based visualization, cannot reveal all the words in a given corpus given typically limited screen resolutions. Significant filtering is necessary. In order to provide for interactive visualization, we carry out several filtering steps at the pre-processing stage. We optionally remove listed *stop words* from the data — words like 'the', 'and' that do not often carry meaning. Domain-specific stop words are identified as the top 0.5 percentile by overall number of documents they occur in, and removed. This captures terms such as 'judge', 'court', and 'circuit' in our data. This filtering is optional because in linguistic study these common words can be very informative if they are unevenly distributed across a corpus.

To further reduce the data size, we identify the word frequency at the 40^{th} percentile when words are sorted ascending by overall occurrence count. We then remove all terms with overall frequency below this cutoff. The 40^{th} percentile was selected to remove much of the *long tail* of terms which are unlikely to be identified as distinguishing — most words removed only occur once or twice in the entire dataset. Our trials have shown that the vast majority of terms with G^2 scores above the significance threshold have frequency > 7. This achieves a vast reduction in the number of terms for which G^2 scores much be calculated at runtime, resulting in a significant speed increase and memory savings with no change to the visualized output.

To reduce the data size further, we also optionally remove words beginning with an upper case letter which do not start a sentence (*initial uppers*). Identifying initial uppers is a quick way to approximate proper noun detection in English. Aside from reducing the data, this technique was necessary to remove place and judge names from the visualization. Initial prototypes revealed that the highest scoring terms were almost exclusively proper nouns. These terms are not informative, as we expect the names of states and cities within a circuit, or the names judges writing decisions in that circuit, to be distinguishing. While this was a useful sanity test on our technique, we removed these terms in the current version. Proper nouns are interesting, however, when viewing the distinguishing terms in the *parties* section of the case data, as common litigants are identified.

6.4.4 *Reverse Stemming*

In order to merge word forms with the same root, such as 'jurisdiction' and 'jurisdictions', we perform stemming using the *Lucene Snowball Stemmer*, an implementation of the Porter stemming algorithm (Porter, 1980). However, the stemming algorithm strips all suffixes, leaving, for example 'jurisdic'. While this is acceptable for counting purposes, we discovered with early prototypes that it is surprisingly difficult to read a text visualization consisting of word stems. As a result, during data pre-processing, we count all occurrences of (word,stem) pairs generated by the stemmer, and retain the most common mapping for each stem. Then, as a final pre-processing step, we reverse the stemming on each term vector using the most common mapping. Thus the visualization shows real words.

As an interesting side-effect, the word forms shown in a Parallel Tag Cloud reveal the most common form of each word within the underlying dataset. We were interested to note that most verbs appear in their past tense form, such as 'averted' and 'insisted', but some appear in present tense, such as 'disagree' and 'want'. By selecting these words in the tag cloud and examining KWIC views for the associated documents, we found a separation between discussion of the facts of a case "The plaintiff averted the problem.", "the district judge erred when she insisted that . . . " and the commentary of the judges "I disagree with my colleagues because . . . ", "We want to reinforce . . . ".

6.4.5 *Visual Variations*

The G^2 score used to identify distinguishing terms provides information about *significant absence* of a word, as well as an unusually high presence. Through graphical tooltips, we provide both positive and negative scores for terms which are present in the tag cloud. However, what if a term is unexpectedly low in a circuit, but does not appear on the tag cloud because it is not high in any other circuit? A tooltip will not help because there is no item to create one for. To address this, we provide a view which selects the top N words per column by the magnitude (absolute value) of the assigned G^2 value. Words are sized and ranked by the absolute value of the score. Negatively scoring terms are distinguished by a red hue. In Figure 6.11 we see 'patent' scores significantly low in all but the Federal and DC Circuits. Perhaps more interestingly, we see 'dissenting'

FIGURE 6.11: In a variation on the visual encoding, we reveal both significantly high and significantly low scores. Lower than expected occurrences appear red, and higher than expected appear black (blue on hover).

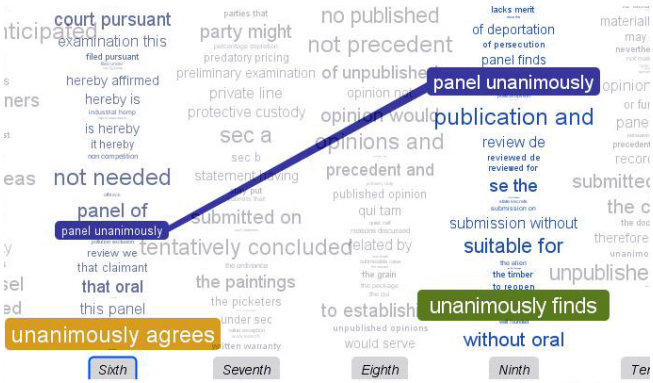


FIGURE 6.12: Parallel Tag Clouds can also be created for two-word phrases.

in the First Circuit, revealing that dissenting opinions are provided in that circuit significantly less often than expected.

Extending our approach to two-word phrases brings several challenges. Tracking multi-word phrases results in an exponential growth in the dataset, and we have more data to fit onto the display space while maintaining legibility. However, we have experimented with two-word phrases using the existing Parallel Tag Clouds implementation, finding differences in verb usage, such as 'unanimously finds' in the Sixth Circuit compared to 'unanimously agrees' in the Ninth (see Figure 6.12).

6.5 IMPLEMENTATION

In order to quickly analyze selected subsets of a large corpus such as the history of the U.S. Circuit Courts of Appeal, significant data preparation is necessary. Our implementation, written in *Java*, makes use of the open-source *Lucene* search engine, both for its search capabilities, and for the {word,count} *term vectors* it stores to support search. In a data pre-processing step we extract parts of each case and pass them to *Lucene* for indexing, stemming, and optional initial upper removal. We also collect the document ID, court ID, and date for each document into a *PostgreSQL* database. In further pre-processing the term vectors are retrieved from *Lucene* for each document in the dataset. A module called the *Term Vector Composer* takes the term vectors for each document, along with the court ID and date, and creates yearly summary vectors of {stem,count[ ]} where count[ ] is an array of counts across each court. The summary

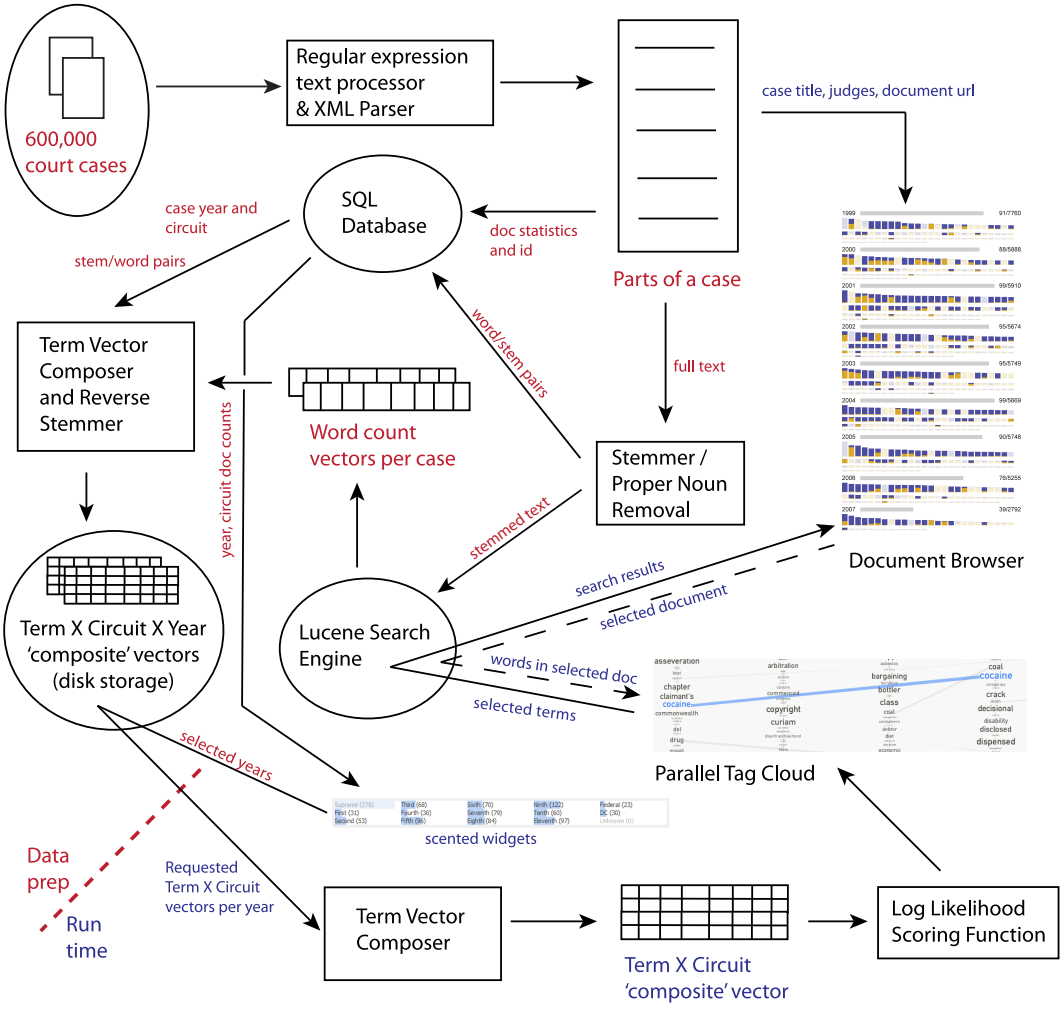


FIGURE 6.13: Data flows through two stages: several preprocessing steps to create aggreagate term vectors in disk storage and a searchable Lucene index (red), and run-time processing to retrieve vectors, calculate scores, manage interaction, and generate the coordinated visualizations (blue).

vectors are filtered to remove stop words, then written to the disk. At runtime, selected year vectors are further composed into a single term vector representing all words in a time range. This is passed to the scoring module and on to the reverse stemmer and visualization. The sequence of data processing operations is illustrated in Figure 6.13.

The indexing and term vector preprocessing operations take approximately ten hours using a 2.53GHz dual core processor with 3GB memory. Preprocessing document term vectors into year vectors reduces the number of composition operations at runtime by a factor of 10,000 (but reduces the time resolution to years). Retrieving and composing term vectors from the disk takes approximately three seconds per year, with the majority of this time spent on disk operations. As even thirty seconds feels like a long wait for a visualization to be populated with a ten year span, after the initial view is presented, we use a background thread to pre-cache term vectors for ten years on either side of a selected time range. When the visualization is closed, the summary vector and the years it contains are saved. If the same time range is later requested, only one load operation is necessary.

6.6 ANALYSIS

As we developed this visualization, we worked with two legal experts. In this section we describe some of the phenomena that were revealed through usage of the system. We do not claim these as original discoveries, but rather as examples of how Parallel Tag Clouds can point to a range of interesting patterns in a real-world data set.

6.6.1 *National vs. Regional Issues*

Because of the arrangement differences between court districts, case law can reveal geographic cultural (and criminal) variations. For instance, drug-related terms appear in most circuits, revealing the national dimension of this problem. Closer inspection, however, uncovers distinct regional flavors: methamphetamine seems to plague midwestern and western states the most, appearing in the Eighth, Ninth, and Tenth circuits. Cocaine cases afflict the East, emerging in the Fourth, Sixth, Seventh, and Eighth circuits. Heroin cases are concentrated in the Second circuit, which

includes New York. These differences might point to either a regional variation in drug use, or perhaps the level of prosecution (see Figure 6.4).

Issues challenging a particular jurisdiction are revealed through the data. For instance, 'deportation' shows up in the Fifth Circuit, which includes Texas, the state with largest crossing border in the U.S., 'gun' appears in the Seventh Circuit, whereas 'copyright' shows up in the Second Circuit, which includes New York. The common occurrence of words, shown through edges in the Parallel Tag Cloud, can reveal similarities as well as differences. For example, in Figure 6.7, we see that the Fourth and Sixth Circuits are similar by virtue of common terms: coal, mining, pneumoconiosis. These similarities make sense since the two circuits are adjacent and share some of the largest coal reserves in the country.

6.6.2 *Language Variation*

Court cases can also provide insight into variations in legal vocabulary and linguistic idiosyncrasies of a particular court. For example, we discovered the odd words 'furculum', 'immurement', and 'impuissant', all in the First Circuit. By revealing the cases in the document browser and isolating the First Circuit, we see that almost all occurrences of these terms originate from a single judge, Judge Selya. A follow up Web search revealed that Judge Selya is well known for his rich and often obscure vocabulary. One legal expert we consulted was fascinated by the potential to use these distinctive pieces of vocabulary as markers to track the influence of a particular judge. For example, the expert pointed to the presence of the word 'ostrich' in the Seventh Circuit. 'Ostrich' here refers to the *ostrich instruction*, shorthand for a particular directive to juries. This term was used almost — but not entirely — exclusively by that circuit over the past ten years. Our legal expert pointed to this as meaningfully idiosyncratic piece of vocabulary that might be used to track attitudes toward jury instructions.

6.6.3 *Forum Shopping*

When bringing a lawsuit, a plaintiff sometimes has a choice between several possible venues. The natural tendency to pick the venue whose judges are historically most favourable to the plaintiff's case is known as *forum shopping*. This phenomenon stands out clearly using our tools. For

example, we can easily see one class of forum shopping by examining data from the years leading up to the creation of the Federal Circuit. The term 'patent' appears for the Seventh Circuit in the period 1970–1980, highly scoring on both occurrence and case-based measures. This is an accurate reflection of legal history: The Federal Circuit was created to combat the varying treatment given to patent rights in the circuit courts; the Seventh Circuit was one of the preferred venues (Crouch, 2006).

6.7 SUMMARY

Parallel Tag Clouds present a method for visualizing differences across facets of a large document corpus. Combined with text mining techniques such as measures of distinguishing terms, this approach can reveal linguistic differences. Within the *space of linguistic visualization*, Parallel Tag Clouds is a visualization method intended for the general public and domain experts to perform content analysis and retrieve documents of interest. It supports interactivity at the data access, data transformation (change scoring function), presentation, and view. The visualization relies on information about the text structure, extracted entities (such as proper nouns), and statistical scoring methods such as G^2. These values are summarized in Figure 6.14.

In upcoming chapters we move away from interactive visualization for content analysis and begin to examine the possibilities of visualizations designed for linguistic, NLP, and computational linguistics (CL) research.

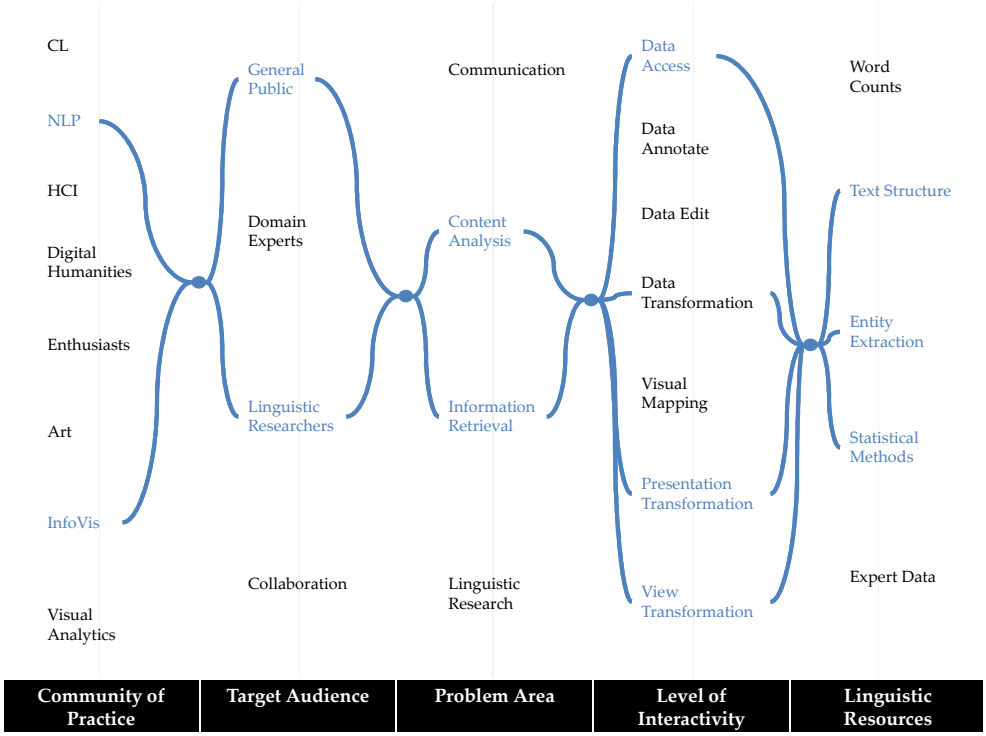


FIGURE 6.14: Parallel Tag Clouds in the space of linguistic visualization.

Part IV

VISUALIZATION FOR LINGUISTIC RESEARCH

7

VISUALIZATION TO REVEAL SET RELATIONS IN MACHINE TRANSLATION RESEARCH

The Milky Way is nothing else but a mass of innumerable stars planted together in clusters.

— Galileo Galilei, 1611

The Milky Way is some more however a mass of the innumerable asterisks, which are planted together in the batteries.

— English-French-German-English machine translation (Tashian, 2009)

While many data sets contain multiple relationships, depicting more than one data relationship within a single visualization is challenging. We introduce this design study as an example of collaboration with natural language processing (NLP) researchers to visualize the multiple relationships within their data.

The resulting algorithms, Bubble Sets, contribute a visualization technique for data that has both a primary data relation with a semantically significant spatial organization and a significant set membership relation in which members of the same set are not necessarily adjacent in the primary layout. In order to maintain the spatial rights of the primary data relation, we avoid layout adjustment techniques that improve set cluster continuity and density. Instead, we use a continuous, possibly concave, isocontour to delineate set membership, without disrupting the primary layout. Optimizations minimize cluster overlap and provide for calculation of the isocontours at interactive speeds. Case studies show how this technique can be used to indicate multiple sets on a variety of common visualizations.

7.1 UNDERSTANDING THE PROBLEM CONTEXT

This investigation was motivated by data analysis challenges faced by NLP researchers working on improvements to their phrase-based machine translation (MT) system. Their problem space is complex: the linguists'

data consists of hundreds of thousands of sentence pairs (same sentence in two languages) and millions of learned translation rules; the translation algorithms run on hundreds of processors in a distributed computing grid; and the team is made up of more than 10 researchers in several locations.

In order to better understand the issues facing our collaborators, we conducted observations and interviews to better understand the domain-specific problems they faced. Preliminary discussions revealed that they spent most of their time sitting at a computer, programming. The data analysis parts of their work — examining training data and testing output — come in unpredictable phases with two weeks or more of coding between cycles. This did not lend itself to long-term observational study, so a series of contextual interviews was conducted in their research environment in order to better understand the work situation.

visualization in-the-wild

Our discussions were wide-ranging and covered many areas of the data, tasks, and work practices: we tried to be as broad as possible, investigating beyond our assumptions of where we thought visualization may be generally helpful. We explored the individual researcher's understanding of their broader research project, their understanding of the algorithms and data, their analysis tools (including ad hoc visualizations, white-board sketches, and notebooks) and practices (using a cognitive walk-through of a typical analysis), their collaboration practices and collaboration support tools, and the ways they measured the success of their research. Our observations hinted at a surprisingly sophisticated reliance on information graphics (custom-made digital graphics and hand drawings) within a research group unfamiliar with the formal concept of information visualization (InfoVis).

To better understand the nuances of the data analysis process, we followed up these discussions with two days of participatory observation (Somekh and Lewin, 2004). In participatory observation, additional insights can be gained through first-hand experience of the data analysis process and context of study. Our collaborators trained me on their typical data analysis tasks for several hours, and then put me to work on data analysis using their existing tools and techniques. The outcomes of my practice data analysis were reviewed with them for validity afterward. Most of the observations were considered valid and noteworthy by the domain experts.

[dev-full:1] Rules Used in 1-Best

ID	Rule	count	gt_prob	green	identity	nonmonotone	missingWord
-10002	CD("14") ↦ "十四"	-	-	1	-	-	-
2728284	NN("border") ↦ "边境"	34	4.463	-	-	-	-
4099120	NNS("achievements") ↦ "成就"	32	4.133	-	-	-	-
18667978	NP-C(NPB(JJ("economic") NN("construction")) PP(IN("in") NP-C(x0:NPB))) ↦ x0 "经济" "建设"	4	6.909	-	-	-	-
19118223	NP-C(NPB(JJ("notable") x0:NNS) PP(IN("in") x1:NP-C)) ↦ x1 x0 "显" "着"	2	7.43	-	-	1	1
41615669	NPB(NPB(NNP("china") POS("'s")) x0:CD JJ("open") x1:NN NNS("cities")) ↦ "中国" x0 "个" x1 "开放" "城市"	1	8.371	-	-	-	1
71090397	S(x0:NP-C) ↦ x0	36	5.713	-	-	-	-
1000000001	TOP(x0:S) ↦ "<foreign-sentence>" x0	-	-	-	-	-	-

TABLE 7.1: An example table of translation rules.

7.1.1 *Analysis Tasks in Phrase-Based Machine Translation*

The task of our translation researcher collaborators is to examine the outputs of the translation system in order to discover any problems in the training data that lead to translation errors. Their technique generates translations by translating segments of a source sentence into fragments of a linguistic parse tree in the target language. The collection of possible parse fragments is then assembled to create a complete parse tree. The translated sentence is read from the leaves through in-order traversal.

This statistical syntax-based machine translation system operates with two probability models: a translation model and a language model. The translation model assigns probabilities that a given sentence segment will translate to a particular parse tree fragment. The language model assigns likelihoods that the candidate translation is a valid sentence in the target language. The two models combine to result in an overall score assigned to a translation candidate. The task of the researchers is to diagnose translation problems that are most often the result of invalid translation pairs in the translation model's training data. To do this, analysts examine hundreds of translation examples, manually tracing problems in the translation back to the source segment and target parse tree fragment pair that caused the problem.

Our NLP collaborators' diagnostic process consists of two steps: [1] review an entire (printed) English parse tree for grammatical or obvious translation problems (see Figure 7.1), [2] if problems are found, scan through several pages of tree fragment-translation pairs to discover problem sources (see Table 7.1). A third data resource, the derivation tree (see Figure 7.2), connects the parse tree and the rules table: rules in the rules

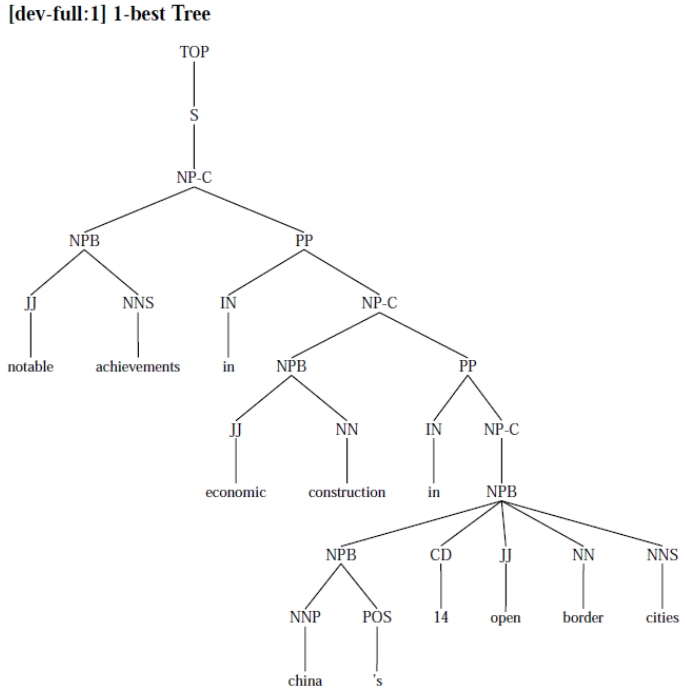


FIGURE 7.1: An example English parse tree.

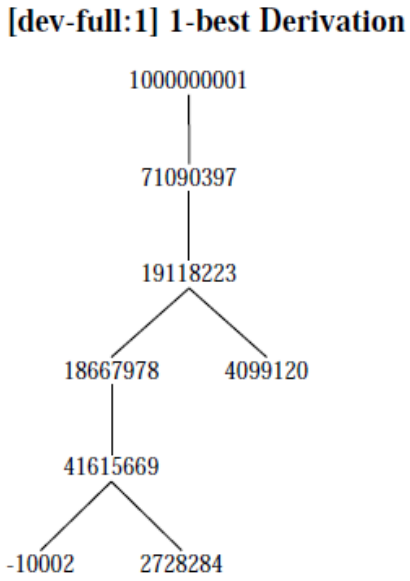


FIGURE 7.2: An example English derivation tree.

table are numbered, these numbers appear in the derivation tree such that if the rules were substituted for their ID numbers, the parse tree would be generated, along with alignment links to the foreign language sentence. In discussions, we found that due to these cumbersome steps, the analysts often skipped the derivation tree and scanned the rules table sequentially for problems.

In their process, there was no way to directly see on the parse tree which nodes corresponded to which rule ID, or which nodes in the parse tree composed a tree fragment (an indivisible unit of translation). Instead, they correlated the parse tree with the derivation tree to discover the rule ID, then inspected the rule in the table for problems. Once a problematic rule was discovered, the derivation tree could be used to trace which nodes of the parse tree were involved (*i. e.*, which nodes in the parse tree were part of the problematic rule). When that set was discovered on their print-outs, analysts often drew bubbles around problematic tree fragments in the parse. Together we saw many opportunities to build an interactive visualization to support this analysis process.

On a high level, the analysis process of our collaborators includes mentally (or manually, through sketching) integrating multiple types of relations within their data: parse tree fragments, derivation sequences, and translation rules. Extrapolating from the process of our collaborators, we observe that many types of data that are important to analysts, such as social network data, geographical data, text data, and statistical data, often contain multiple types of relationships. These can include *set relations* that group many data items into a category, *connection relations* amongst pairs of data items, *ordered relations*, *quantitative relations*, and *spatially explicit* relations such as positions on geographic maps. Common approaches to visualizing set data focus on solutions that integrate clustering and bounding outlines. That is, when possible, set members are moved into close proximity contained within a convex hull. When set member proximity is not possible, alternate approaches make use of additional visual attributes, such as colour, symbol, or texture, to indicate that discontinuous items or groups are in fact members of the same set. With the translation parse tree analysis task, we have a structured relation (the parse) and other relations (tree fragment membership) which may not necessarily correspond to node proximity in the tree. Therefore, we designed the Bubble Sets visualization to provide continuous bounding contours, creating outlines analogous to the hand-drawn enclosures without requiring spatial reorganizing.

Before describing the details of our solution and its application to the analysis challeges of our MT collaborators, we first review some background related to multiple-relation visualizations. We advocate generally for careful consideration of *spatial rights* when visualizing multiple relations and review the related work using spatial rights as a means to differentiate between the types of visualizations that contain both connection and set relations. Then we outline our algorithmic approach to determining and rendering set boundaries, and illustrate our technique with four case studies.

7.2 BACKGROUND

We use the common definition of *set*: a collection of unordered items, possibly empty, with no repeated values. This includes Freiler *et al.*'s (2008) definition of set membership by *set-type attribute*. Also, we include sets that are defined implicitly by relationships amongst members (a group of friends), sets which share data attribute values (all cars with air conditioning), and sets which are arbitrarily specified (personally-selected).

One approach to set visualization is to consider the set relation as primary and create a spatial layout according to set membership (Freiler *et al.*, 2008). In this situation spatial proximity within sets can be achieved. Another approach is to spatially adjust or re-cluster a given visualization to bring set members into closer proximity, making it possible to visually group them with a convex hull. Since both of the above approaches result in spatially clustered set members, visually containing them in convex hulls can be effective (Dwyer *et al.*, 2008; Heer and danah boyd, 2005; Perer and Shneiderman, 2006). Convex hulls are fast to calculate and well-suited to cohesive clusters which are separated from neighbouring groups. However, if the layout contains items that are not set members but are within the set region spatially, these items appear within the convex region determined by set members, and thus will appear to be set members. This is seen in *ScatterDice* (Elmqvist *et al.*, 2008) when scatter plot axes change after sets are defined with the lasso tool.

Additionally, in some spatially assigned layouts, such as scatter plots, and in other data representations, such as maps, the semantics of the layout preclude spatial re-positioning. In these situations use of convex hulls to encircle set members is not effective. Here set membership is

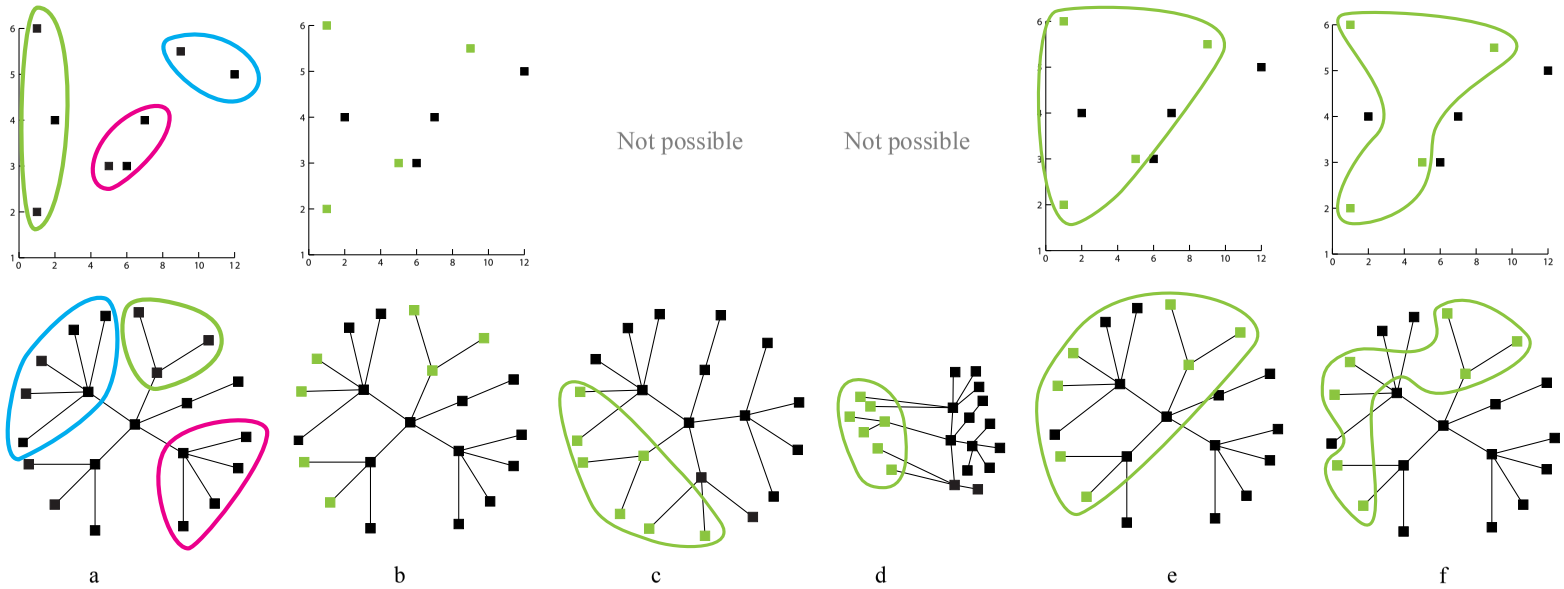


FIGURE 7.3: Rows — Set relations over statistical plots (top row) and node-link graphs (bottom row). Columns — (a) Simultaneous spatial rights: set membership is based on proximity in layout. (b) Spatial rights for the connection relation: set members indicated through common item colour. (c) Hybrid spatial rights: layout based on connection relation is adjusted to bring set members into closer proximity. (d) Spatial rights for set relation: layout brings set members into a tight cluster, obscuring the structure of the connection relation. (e–f) Spatial rights for attribute-based layouts and connection relations: the set relation is drawn atop existing layouts using either the traditional convex hull (e) or Bubble Sets (implicit surfaces) (f).

sometimes indicated through discontinuous set outlines and/or the use of colour and symbols. To provide a method that can visually contain set members within an outline, we calculate a polymorph hull which can have convex and concave regions. This hull can avoid including items which are not set members, except in very dense arrangements where a human would also have difficulty manually drawing the enclosure. We use implicit surfaces similar to those used by Watanabe *et al.* (2007) and apply them to create contiguous multi-set visualizations over multi-relational information visualizations. The requirement for more complex hull outlines to support set visualization within many different types of visual representations is closely tied to the spatial semantics and the concept of spatial rights.

The requirement for more complex hull outlines to support set visualization within many different types of visual representations is closely tied to the spatial semantics and the concept of spatial rights.

7.2.1 *Assigning Spatial Rights*

Since research into how we perceive visualizations indicates that the spatial positioning of data items may be the most salient of the possible visual encodings such as position, colour, shape, *etc.* (see §2.3.3), it may well be important to preserve positioning of data items while providing set membership visual containers. We refer to this concept of granting primacy to the particular aspect of the data on which the layout is based as *spatial rights* (see §2.3.6).

Social network data exemplify the different types of relations we consider. These sorts of rich data sources are important to understanding organizations, online culture, and computer-based collaboration. A connection relation may be that friends are directly connected to one another, or that supervisors are connected to their direct reports in a management hierarchy. A set relation could be defined in several ways: based on graph characteristics such as cliques of closely connected friends, based on demographic information such as occupation or level of education, or arbitrarily determined, for example, by an analyst. *Set* relations are sometimes called *categorical* relations, where all items sharing a value for a categorical data attribute comprise a set. Finally, additional ordered and quantitative attributes may be present, and could be used to annotate or position the visual items, such as arranging by date of birth or annual

income. With these data, it is likely most intuitive to provide a layout based on the connectedness information, as that is the 'network' in social network analysis. In this case, it may not be possible or desirable to adjust the layout to maximize set contiguity or density. Doing so may disrupt the meaning provided by the connection-based layout. Traditional set visualization techniques will not be adequate, and our approach may be more applicable.

Traditionally, either the connection relation or the set relation is afforded primacy, and the spatial layout of a visualization is based on that relation. There is a trade-off in design — optimized layouts for connections may lead to unclear sets; optimized layouts for sets may lead to confusing edge crossings and a loss of visible structure. An alternative is to design a coordinated view system, in which one view is based on the set relation, and the other is based on the connection relation. However, in this work we will restrict ourselves to tightly coupled relations in a single-view visualization. Our implicit surface solution clarifies set membership while not disrupting information-carrying layouts based on connection relations and ordered attributes. We will explore the related work by defining five configurations of spatial rights and giving examples from the literature where possible. Figure 7.3 illustrates this discussion by showing in diagrams different ways to indicate set membership in scatterplots and on tree layouts.

Simultaneous Spatial Rights

If a dataset contains one set relation and one connection relation, it may be spatialized based on either. One may desire contiguous, compact clusters to represent sets, or one may wish to design a layout to minimize overlap of edges in the connection relation. These two goals are not always mutually exclusive. In some cases, set membership is a direct function of the connectedness characteristics of a graph, or even the spatial proximity resulting from a layout based on graph structure. In these cases the desired layout can be created based on the connection relation, and sets will appear closely clustered and contiguous, giving both spatial rights. Figure 7.3a shows a scatter plot (top) and a tree (bottom) where each set's spatial proximity allows for bounding by a convex hull. Examples of this include the friend clusters in *Vizster* (Heer and danah boyd, 2005) or spatially-determined aggregates in level-of-detail visualizations (Balzer and Deussen, 2007). Our investigation of the design space of multi-

relationship visualizations reveals that assigning simultaneous spatial rights is not always possible.

Hybrid Spatial Rights

The set relation may be pre-determined by data characteristics or set membership may be arbitrarily selected by an analyst. In these cases the layout may be adjusted to bring set members closer together. In this condition, the connection relation and set relation share spatial rights in a hybrid layout. However, reorganizing a layout to create more continuous and denser sets may have serious consequences for the readability of the primary connection relation. For example, nodes in a tree may have a meaningful order which reorganization may disrupt. In Figure 7.3b for both the tree and scatterplots set membership is discontinuous and is indicated by node colouring. In Figure 7.3c the tree from Figure 7.3b is re-organized to provide set proximity, however, this re-organization would destroy meaning in the scatterplot. While we do not know of examples which explicitly create a hybrid layout for set and connection relations, Phan *et al.* (2005) report a hybrid layout based on hierarchical clustering and adjusted to provide node separation. Dwyer *et al.* (2008) use a hybrid layout technique to provide for fast calculation of large-scale overviews which are adjusted in detailed views for higher-quality layout.

Spatial Rights for Set Relations

When the set relation is the primary relation, it can be assigned spatial rights. Items are then arranged to maximize set separation, continuity, and density. Connection relations are drawn atop this layout (Figure 7.3d), however, this approach does not apply to scatterplots. We do not know of any research that affords primary spatial rights to a set relation while drawing a connection relation atop. The closest analogues are approaches using multiple connection relations, such as using hierarchical data to lay a treemap out, then drawing additional connection relations atop (Fekete *et al.*, 2003).

Spatial Rights for Connection Relation

When the connection relation is the primary relation, it is best to assign it spatial rights. When set membership is not based on the connection relation, but rather on an unrelated data attribute, or even interactive

selection, common visualization techniques such as convex hulls are not sufficient (Figure 7.3e). Visualizing set relations for data items in a predetermined visual layout is often difficult due to spatial discontinuities, and set overlaps. Set-membership ambiguities can be introduced by the use of convex hull algorithms. Noting set membership with another visual encoding such as symbols or colours on the data items is helpful but lacks the clarity of a single enclosure. Indicating set membership with 'bubbles' — contours that tightly wrap set members — has been attempted in the past, however, this approach only offered continuous set membership enclosure for proximal groupings (Heine and Scheuermann, 2007; Watanabe *et al.*, 2007). Byelas and Telea (2006) provide connected set enclosure visualizations for UML using outer skeleton construction and handle incorrectly included items by second pass cutouts. In this work, we introduce an efficient implementation of implicit surfaces which allows for contiguous sets to be drawn over arbitrary layouts, while reducing set membership ambiguity problems using interactive highlighting (Figure 7.3f).

Spatial Rights Explicit in Data

With some data types such as maps the spatial rights are explicit in the data semantics. Here set memberships can exist across multiple distances such as the sets of all capital cities or the cities with populations over a million. With this type of data spatial re-organizing is not an option and solutions such our approach diagrammed in Figure 7.3f) are essential.

7.2.2 *Implicit Surfaces*

While implicit surfaces have been previously used to illustrate set relations over graphical objects (Heine and Scheuermann, 2007; Watanabe *et al.*, 2007), in both works, the set relation is defined by the spatial proximity of graphical objects — increasing an item's distance from the set centroid removes the item from the set; pushing an item toward the set centroid adds it. As we aim to maintain the visualization of set membership for any spatial relationship between set members, dragging an item away from a set centroid will not remove it from the set. Thus we provide alternative interaction techniques to modify set membership. Watanabe *et al.* provide no option for set members existing across long distances. Heine and Scheuermann allow for a single set to be divided into multiple

subgroups separated by distances by using bubbles of the same colour. They do not detect an isocontour, rather using pixel-based shading to display sets. Thus rich interaction with the set is not possible. In our work, we maintain a continuous and connected contour around all set members irrespective of distance or spatial organization.

7.3 ALGORITHMS

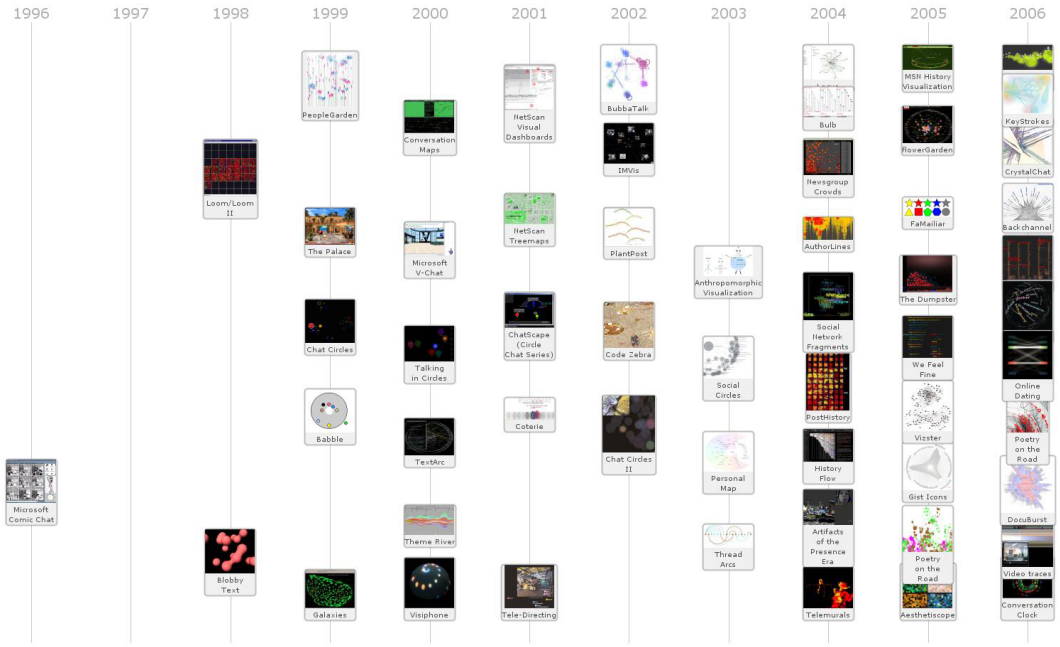
Our approach arose from observing curved and complex boundaries hand-drawn by people to indicate set relations. To simulate these natural-looking boundaries, our method requires: (a) all set members to be enclosed, (b) non-members to be excluded, (c) where non-members occur within boundaries, visual and interactive hints to clarify membership, (d) rendering to allow for interactive adjustment.

Implicit surfaces, more accurately called *implicit contours* in 2D, are well suited to address these requirements (Blinn, 1982). In this section we will describe our version of implicit contours (*bubbles*) and the heuristics we employ to fulfill the requirements and create accurate and aesthetically pleasing bubbles around set members. A high level overview of the steps to creating Bubble Sets is outlined in Figure 7.4. In the remainder of this section we will elaborate on the steps in the process.

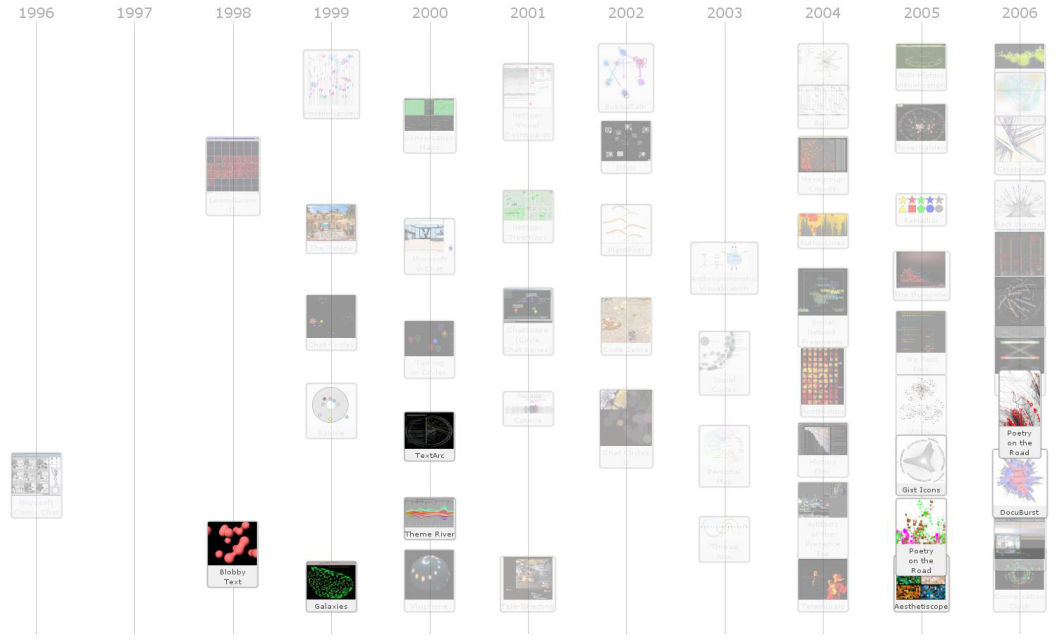
7.3.1 *Surface Routing*

Before drawing an implicit contour around set members, we first create an invisible skeleton of connections between set members. We call these connections *virtual edges* and their structure forms a basis for contour routing between members and around obstacles, allowing set boundaries to flow in an aesthetically pleasing way while avoiding overlaps with nodes and maintaining set connectivity.

First, for each set, we identify the set members. Using the locations of the set members, we then define an *active region* as the rectangular bounding box which includes all set members, increased on all sides by a buffer of R_1. Because R_1 is the maximum distance of the set boundary to any item in the set, only items in the active region can be close enough to set members to affect the bubble creation. Therefore, for speed purposes,

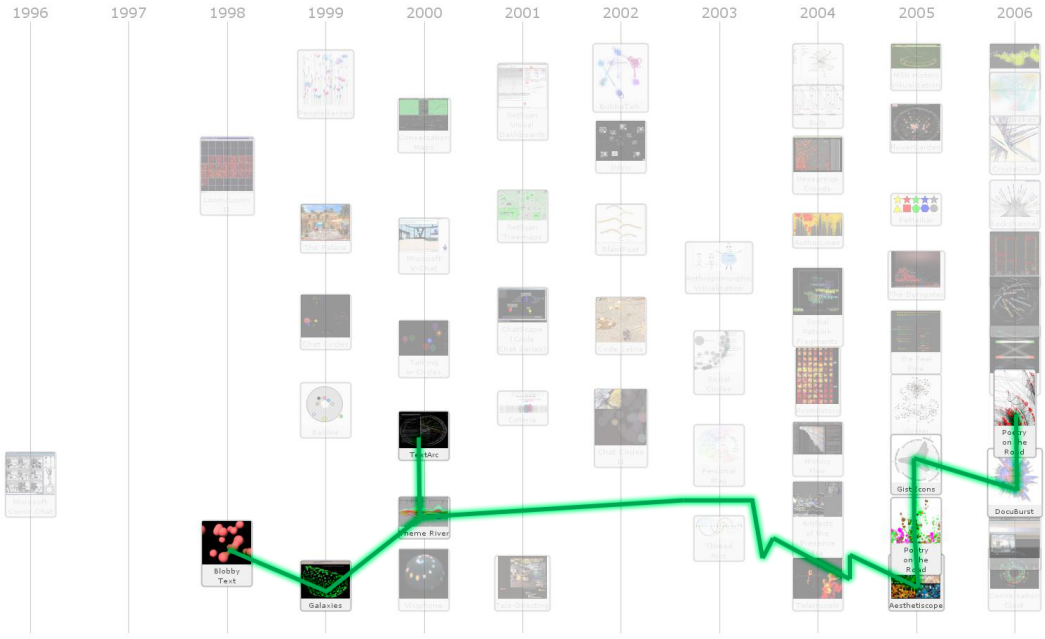


(a) Primary layout: The items in a visualization are positioned according to the pre-determined primary spatial layout.

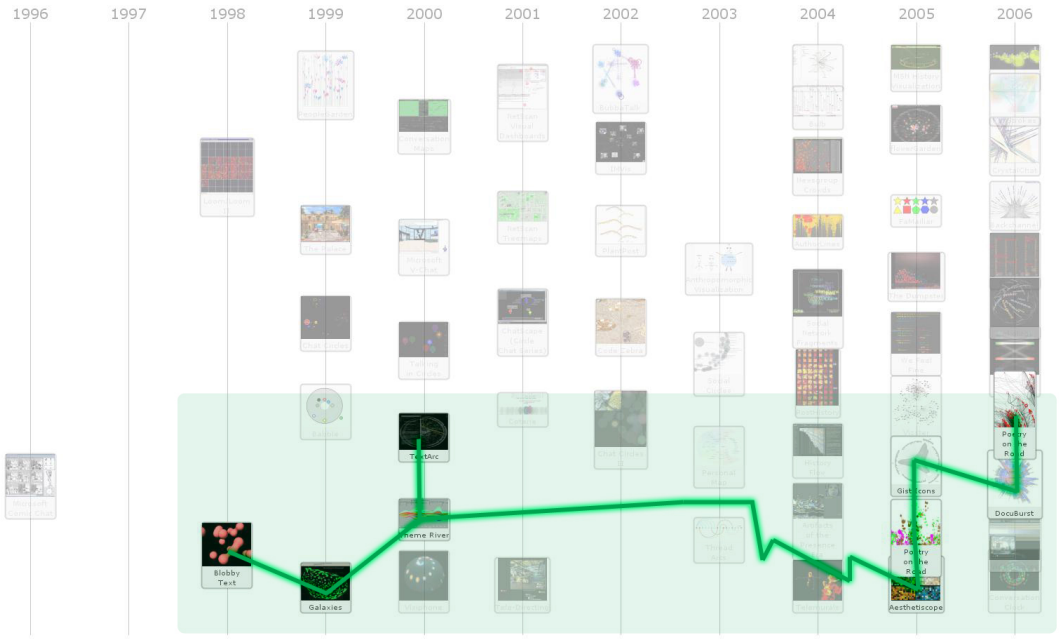


(b) Identify set members: In this image set members are distinguished from non-set members by opacity. Set membership is not based on proximity, but rather relations within the data that are not otherwise spatialized.

FIGURE 7.4: Process overview for building Bubble Sets.

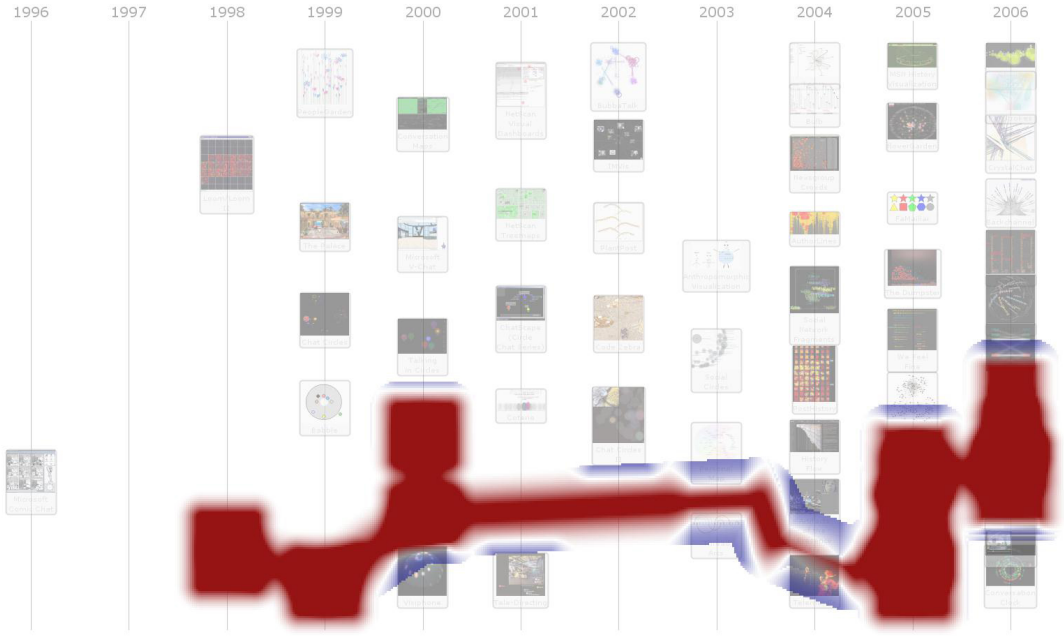


(c) Determine the virtual edges: Set members are connected by virtual edges, which are routed around obstacles. Virtual edges are shown here in green.

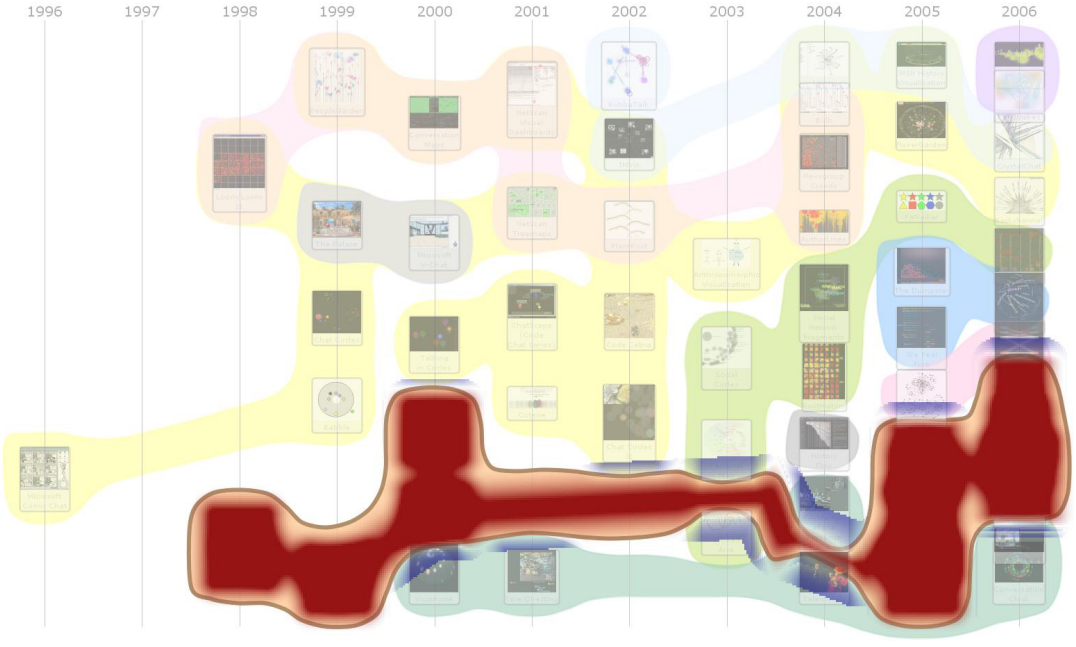


(d) Determine active region: The smallest area which is guaranteed to contain all items in the set is determined using a bounding box and buffer of R_1.

FIGURE 7.4: Process overview for building Bubble Sets (continued).



(e) Calculate energy field: Items in the set contribute positive energy (red), items outside the set contribute negative energy (blue). The energy field within the active region is visualized here for illustration.



(f) Determine set boundary: Using the marching squares algorithm, an iso-contour is determined where the energy equals a set threshold. After a continuous contour is determined, all set items are tested for containment within the contour. If any test fails, energy parameters are adjusted and the energy and contour calculations are repeated.

FIGURE 7.4: Process overview for building Bubble Sets (continued).

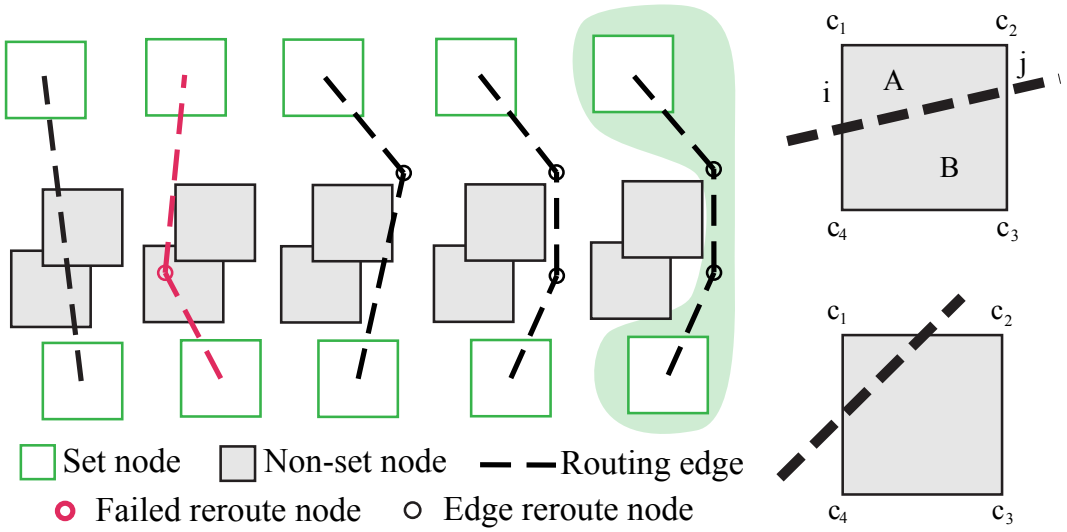


FIGURE 7.5: Contour connectedness is assured through virtual edges which add to the energy distribution for the set. (l-r): A virtual edge passing through a node is detected. A new control point is created at a corner of the obstacle's bounding box and the test repeated. As the test fails, the diagonally opposite corner is then used and no obstacle is found. Additional control points are created at the corners to route edge around the obstacle. The final set of virtual edges contributes to the contour calculation, allowing the set contour to avoid the obstacles and remain connected. Far right: the configurations for creating virtual nodes with Algorithm 7.1.

items and pixels outside the active region are not included in the energy calculations.

There are two options for routing the bubble surface. First, if *structural edges* (edges that are part of the data) are included, the bubble surface should follow them. For example, if a set over a node-link graph includes both nodes and edges, we simply use these items to calculate the bubble surface. However, as discussed previously, set relations may not have any dependence on a connection relation in the data. In these cases, we ignore any edge structure in the visualization and determine bubble routing based on the constraint that bubbles must, where possible, avoid overlapping or including non-set members within the bubble boundary. To achieve this, we route the bubble surface around items which should not be enclosed using an invisible backbone of edges that connect set members while avoiding non-included items. This backbone is initialized by iterating through set members and connecting them to an optimal already-visited member of the same set using a straight line between item centers. The optimal neighbour $j_{optimal}$ for node i is selected to minimize

the function $cost(j) = distance(i,j) * obstacles(i,j)$ where $obstacles(i,j)$ is the count of non-set members on the direct path between i and j. This function balances a preference for close connections with the simplicity of straighter paths. We start this operation with the item nearest the set centroid and proceed outward. This encourages 'blob' shapes rather than 'snake' shapes (Figure 7.6). This process ensures all set members are connected.

Extending the edge routing algorithm of flow maps (Phan *et al.*, 2005), we then test all non-set members within the active region for intersection with the virtual edges. If an intersection is found, we split the edge and route it around the blocking node by creating a new control point offset from one of the node corners and connecting the original edge endpoints to this control point, creating 2 virtual edges. We place the control point R_1 away from the corner to provide for a buffer around the blocking item. If the creation of the new control point initially fails, we iteratively try routing around other corners of the obstacle and reducing the buffer. This process, detailed in Figure 7.5 and Algorithm 7.1, is repeated for the new segments until an iteration limit is reached or no intersections are found. This method does not work for the case where a set member's bounds are completely contained within a non-set member (see scatter plot case study). Our algorithm is $O(KN(N+HW))$ for K sets, N items, and a pixel field of H by W. However heuristics such as restricting energy calculations to the active region around an item reduce the average case considerably.

An implicit surface is simply a contour such that the energy $E(x,y) = c$ where c is a constant energy value. We consider the display space to be a grid upon which we calculate *energy* values for each cell (pixel). When the energy function is continuous, continuous contours are guaranteed. However, there may be more than one separate contour in the energy grid. Contours are defined by the presence of items on the grid. An item is an object that may or may not be a set member. For each pixel the energy is the sum of influences of nearby items, as a function of distance:

$$s_{pixel} = \{j | j \in items, distance_{j,pixel} < R_1\} \quad (7.1)$$

$$energy(pixel) = \sum_{i \in s_{pixel}} w_i (R_1 - distance_{i,pixel})^2 / (R_1 - R_0)^2 \quad (7.2)$$

Algorithm 7.1 Route virtual edges around obstacles (see Figure 7.5).

```
while ∃ virtual edge l_{k,m} which intersects obstacle do
  n ← null, swap ← false
  while (n = null ∨ n intersects obstacle) ∧ (buffer > 0) do
    if l_{k,m} intersects adjacent edges of obstacle bounds then
      add virtual node n at corner of adjacent edges
    else
      if Area(A) ≤ Area(B) then
        if i > j then
          add virtual node n at (swap ? c_1 : c_3) + buffer
        else
          add virtual node n at (swap ? c_2 : c_4) + buffer
        end if
      else
        if i > j then
          add virtual node n at (swap ? c_3 : c_1) + buffer
        else
          add virtual node n at (swap ? c_4 : c_2) + buffer
        end if
      end if
    end if
    if swap then
      reduce buffer
    end if
    swap ← ¬swap
  end while
  split virtual edge into l_{k,n} and l_{n,m}
  reset buffer
end while
```

where R_0 is the distance at which energy is 1, R_1 is the distance at which energy reaches 0, w is the weight assigned to the item, *items* is the set of all items in the space, and s_{pixel} is the set of influencing items within R_1 of the pixel. An isolated point item, then, will have a circular isocontour at radius R_0 and an energy field extent of R_1. As items are often not points, but rather shapes such as rectangles, we use Euclidean distance to the nearest point on the shape surface. Inside shapes we assign $distance = 0$. As the energy function reaches its root at R_1, only items within R_1 of a pixel are included in the energy calculation and will have a non-zero effect. After calculating energy values for all grid cells, we use a 2D version of marching cubes (Lorensen and Cline, 1987) to trace the contour. As described in the following sections, we adapted this general method with additional steps to ensure sets are connected and contain all set

Algorithm 7.2 Determining a bubble set boundary.

given items with positions
for all sets $s \in S$ **do**
  find centroid c of s
  for all items $i \in s$, order ascending by distance to c **do**
    find optimal neighbour $j \in s$
    find best route from i to j
    for all cells (pixel or pixel group) within R_1 of i **do**
      add energy due to i
      add energy due to nearest virtual edge $i \rightarrow j$
      subtract energy due to nearby non-set members $k \notin s$
    end for
  end for
  repeat
    perform marching squares to discover isocontour $\bar{s}$
    reduce threshold
  until $\forall i \in s$, isocontour $\bar{s}$ contains(i)
  draw cardinal splines using every Nth point on the contour
end for

members, while excluding items not in the set. The simplified algorithm for calculating bubble boundaries is presented as Algorithm 7.2.

7.3.2 *Label Placement*

If the data contains a label for the Bubble Set, it is placed along the longest virtual edge in the set (see Figure 7.8). If there are no virtual edges long enough to accommodate the label at the minimum font size, the label is drawn above the first item in the set. Labels can be placed before or after energy calculations. If placed before energy calculations (the default), the label is treated as a set member item, ensuring the set boundary includes the label bounds.

7.3.3 *Energy Calculation*

To calculate the energy field, several techniques are employed to gain speed. First, the display space is divided into square pixel groups which are treated as a single pixel. While this lowers the resolution of the surface calculation, the visual artifacts introduced are minimal, and it actually has the effect of smoothing the bubble surface. We dynamically adjust the pixel group size to provide interactivity when items are dragged (pixel

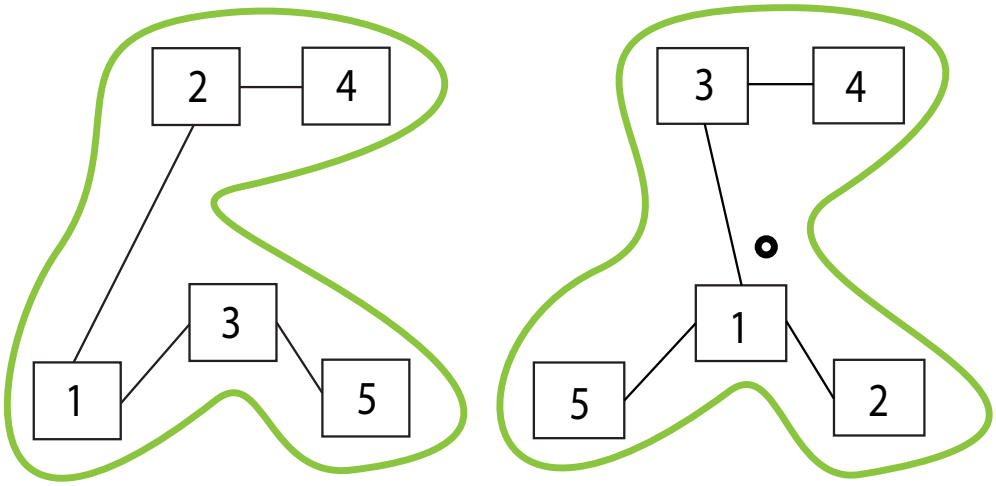


FIGURE 7.6: The order of connecting set members with virtual edges affects the generated shape. Left-to-right, top-to-bottom connection generates a snake-like virtual edge configuration (left), while connecting from the centroid (black circle) outward generates a more blobby shape (right).

group 9×9) and higher quality rendering when the scene is static (pixel group 3×3).

The energy field is calculated for each set in sequence. Nodes and edges (both structural and virtual) in the set are given a positive weight of 1. Negative energy influences result in the implicit surface being pushed away from items not included in the set. Nodes not included in the set are weighted -0.8. Non-set edges are usually given a weight of 0, as bubbles will not be able to avoid edge crossings, and energy reductions at edge crossings can cause surface discontinuities. The energy contribution of visual items is also dependent on R_0, and R_1 — these parameters must be tuned for a particular application depending on the size and spacing of items, and the bubble margins desired.

For a given set, we first calculate the positive energy influences for each pixel group in the active region. That is, for all set members and virtual edge sets, we calculate energy contributions for all pixel groups within R_1 of the item. For a given pixel group and virtual edge set (route between two nodes), only the edge segment closest to the pixel group contributes energy. This avoids overly large positive energy values near segment connection points, which would lead to bulges in the final surface. Next, for pixel groups with total energy greater than zero, we add the negative influence of entities which are not in the set. As regions of zero or negative

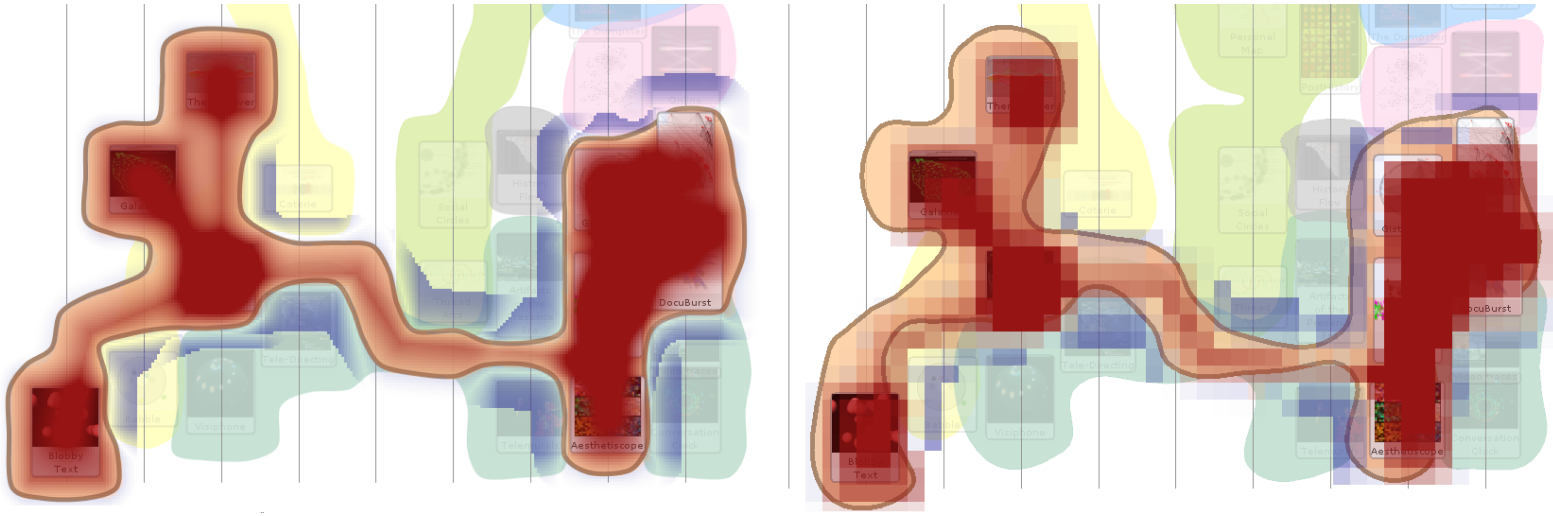


FIGURE 7.7: As visualized here, the energy field for a single set can be calculated at varying levels of precision. At left, with pixel group = 1 for high quality rendering, at right with pixel group = 9 for interactive animation. Non-member items and additional sets are faded in the background. Red areas indicate positive energy, blue areas are negative energy. Note the visible bends in the virtual edges route in the center and far left, an effect of the routing algorithm. The isocontour is shown as a solid brown boundary.

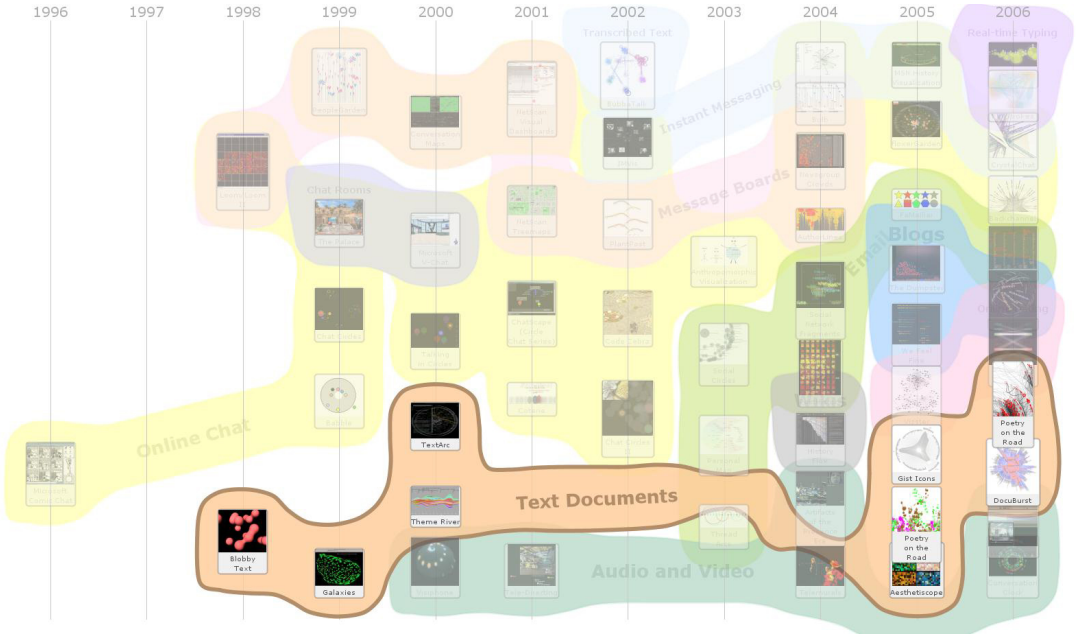


FIGURE 7.8: If a Bubble Set has an associated label, it can be drawn along the longest unobstructed virtual edge. If no such edge will fit the label at a minimum font size, the label is drawn across the top of node. Labels can be optionally included as set items in the energy calculation, guaranteeing containment within the Bubble Set contour.

energy will never be part of the isocontour, we do not calculate negative energy contributions unless the pixel group already has a positive energy. This provides a significant reduction in the time required to fill the energy surface of the active region. A visualization of the energy field underlying a set is shown in Figure 7.7.

7.3.4 *Contour Discovery*

We used a 2D version of marching cubes (Lorensen and Cline, 1987) to discover an isocontour for each set. After discovering the isocontour, we check all set members to ensure their centers are within the contour enclosure. If they are not, this indicates a disconnection. In this case, we iteratively reduce the energy threshold by a factor *alpha*, repeating marching squares until all members are included. In very dense layouts, it may be necessary for the set contour to pass through items which are not included in the set, for example if items are adjacent without space for routing around. In practice, such situations would be difficult for

a person to draw — if no space is available for the set to route around an obstacle, our algorithm will eventually go through it. The marching squares step is fast to repeat: $O(H + W)$ for an active region of H by W pixel groups. If, however, after N iterations the set is still disconnected, we manipulate the energy field, a slower process ($O(HWK)$ for K items in the active region). We increase the positive weights by a factor β, decrease the negative weights by a factor γ and recalculate the energy field, followed by another iteration of marching squares. We repeat the energy manipulation for N additional iterations or until the set boundary encloses all member items.

Negative energy contributions serve to 'push' the set boundary away from non-set members near the contour boundary. Surface routing attempts to work the boundary around obstacles, but when a dense set encloses a non-set member, the algorithm may fail to exclude that item. If a non-set member is discovered within the contour, we highlight it with a white border. It would also be possible to create a hole using an iteration of marching squares beginning at the center of the non-set member and working outward to discover an interior contour. We leave this for future work and use highlighting and interactivity to clarify such occurrences.

To render the surface we use cardinal splines, using every L^{th} point on the surface discovered by marching squares as a control point. The selection of L involves a trade-off between smoothing and precision, and is dependent also on the pixel group size. We lean toward smoother surfaces and select $L = 10$ in the examples that follow. We colour the interiors of the surfaces with a transparent version of the border colour to more clearly indicate the extent of the set region.

7.3.5 *Interaction*

Interactions such as adding items to sets, removing items from sets, creating new sets and moving items within sets are provided. Since proximity does not determine set membership, we use moded interaction. First a set is made active with a right-click, then individual items can be added or removed by clicking them. When complete, the set is deactivated and its new contents are fixed. Depending on the application requirements, it is also possible to select and move an entire set by clicking on the set background. Our requirement for surface calculation and rendering at interactive speeds is met, as discussed, by defining active regions, pixel

groupings, and only calculating negative influence for pixels with energy greater than zero. Sets are rendered from largest to smallest to facilitate picking, ensuring smaller sets are not completely covered. After the initial rendering, we recalculate only the contours for sets with changing configurations. Specifically, for any moved item i, we recalculate only the surfaces for sets $s \in S$ if i is in the active region of s. The speed gain of this heuristic is dependent on the density of the display space and the number of sets.

We cannot guarantee that non-set member items will be excluded from the set boundary in all cases. The use of surface routing and negative energy influences minimizes this occurrence, but it remains possible depending on the density of the graph and the particular layout of items. To clarify set membership, we visually separate overlapping or enclosed non-set items from the set using a white border. Additionally, when a set member or set boundary is under the mouse, all non-set member items are faded out (made partially transparent).

7.4 BUBBLE SETS OVER MACHINE TRANSLATION PARSE TREES

Returning to our collaboration with the NLP researchers, the data for the visualization consisted of a connection relation (parse tree) and a set relation indicating which tree fragments were a unit of translation. We assigned spatial rights to the connection relation, laying out the parse tree with an improved version of Reingold and Tilford's layout (Buchheim *et al.*, 2002). The set relations are not determined by proximity in the parse tree, but rather are specified by the way in which the tree is constructed from fragments. However, the set relation is not completely independent of tree structure — individual sets are guaranteed to be connected through tree edges. Thus, the structural edges rather than our virtual edges are used for surface routing.

Our set visualization approach is well-suited to reveal the tree fragments directly on the output parse tree. Set background hue is selected on the basis of the category of the fragment (a classification important to the analysts), and the background transparency is based on the confidence score assigned to the set by the translation algorithm (darker means more confident) (Figure 7.9). Traditional methods such as convex hulls would not suit this application, as they are unable to exclude non-set members (see Figure 7.10). As part of a tool for analyzing translation models, our

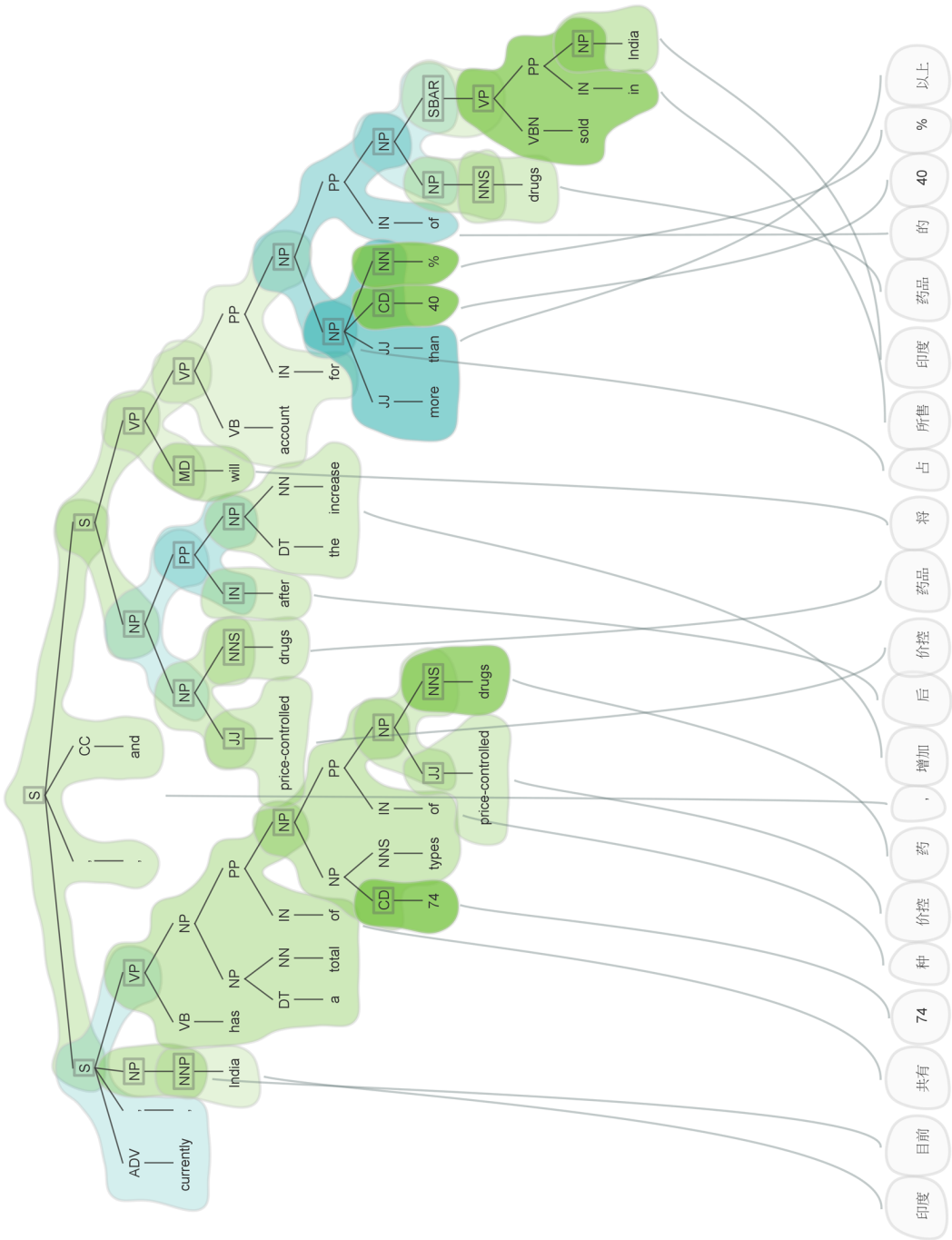
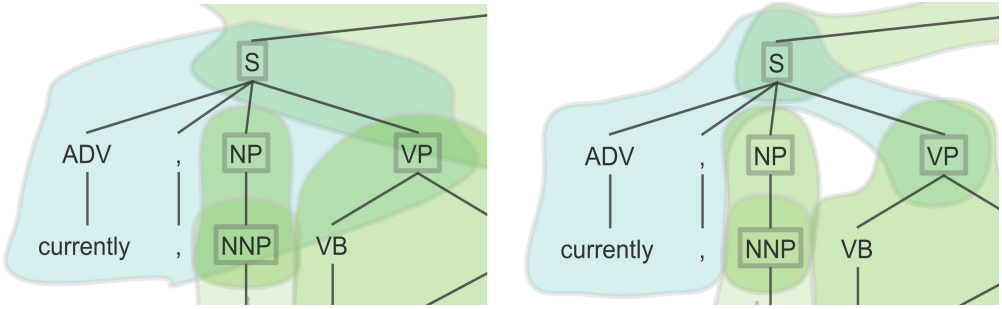


FIGURE 7.9: An example of a translation parse tree from Chinese to English. Bubble colour indicates the type of translation rule, increasing bubble opacity indicates a higher probability score. Edges indicate correspondence between sets over the parse tree and Chinese characters.



(a) (b)

FIGURE 7.10: Contrasting surface drawing techniques: (a) The blue convex hull includes non-set members within the boundary, whereas the corresponding Bubble Set (b) does not.

visualization allows translation researchers to review translation parse trees for problems and annotate discoveries directly on the visualization, without the need for lengthy tables of translation tree fragments.

7.5 GENERALIZING BUBBLE SETS

Bubble Sets were originally developed for MT parse trees, but can generally be applied over any exisiting 2D visualization for which set membership visualization is desired. In the remainder of this section, we will describe additional case studies using Bubble Sets, showing the versatility of this approach.

7.5.1 *Research Articles Timeline*

Spatial tools to organize personal archives of PDF documents have been reported, including the *Dynapad* tool which provides both clumped (set-based) and timeline (attribute-value-based) layouts (Bauer *et al.*, 2005). However, *Dynapad* is not able to display the set relation when the PDF article icons are displayed in the timeline formation. Combined timeline and set views of the research literature are used for personal information organization, and for communicating an organization applied to a set of articles (Tat, 2007, p. 23). Several people were involved with the discussion about which articles belonged in each set, and the groupings were revised several times. Inspired by this activity and the manually-created

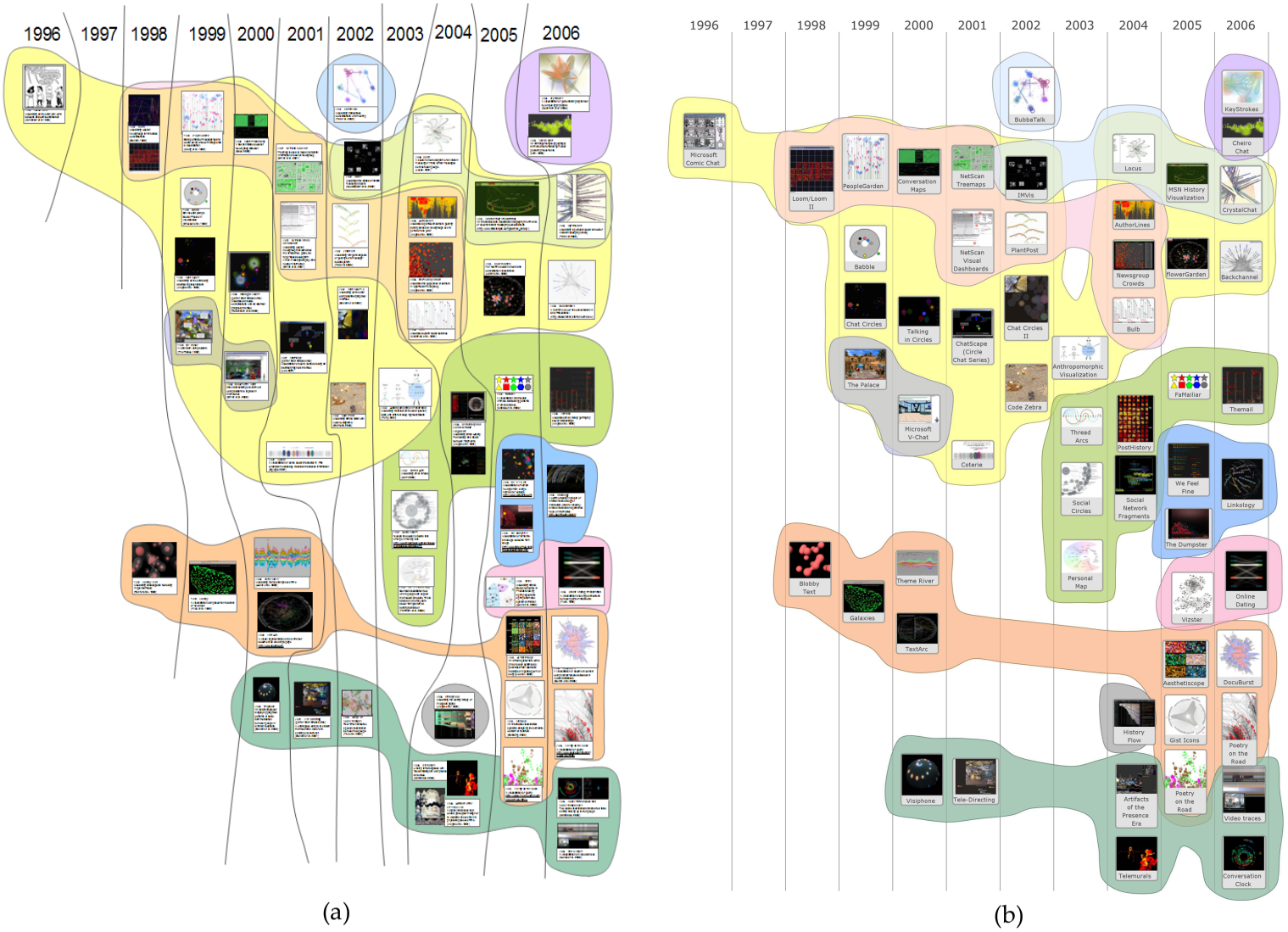


FIGURE 7.11: Grouping research articles on a timeline. (a) Manually-created sketch (courtesy Tat (2007)). (b) Bubble Sets visualization of the same data.

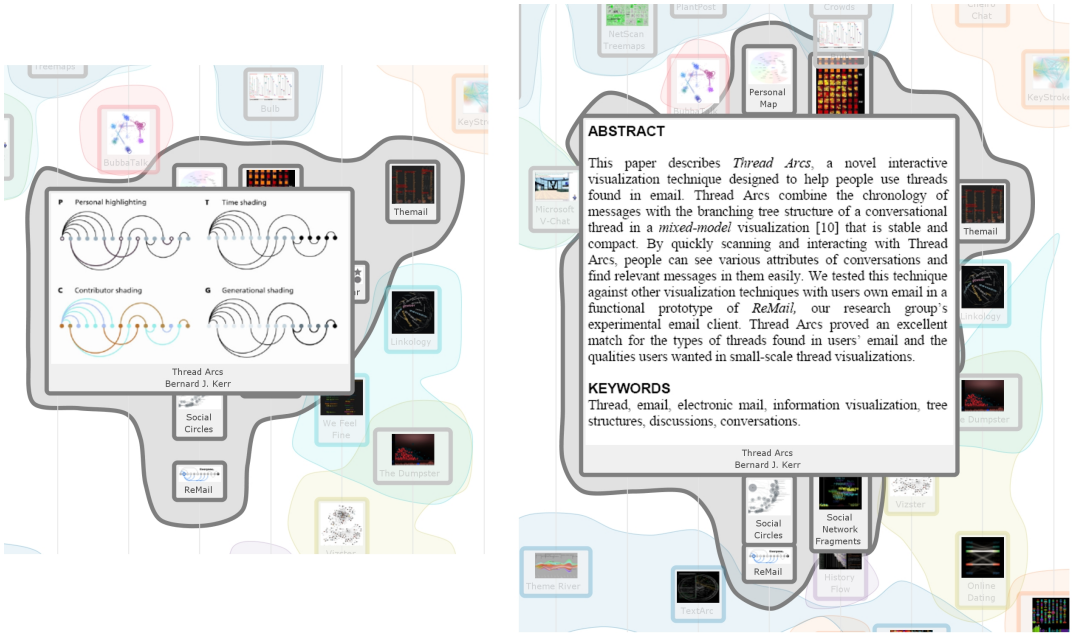


FIGURE 7.12: Items can be expanded to reveal a larger image or the article's abstract. The boundary moves to accommodate the larger item, and other items move along the y-axis to remain visible and selectable.

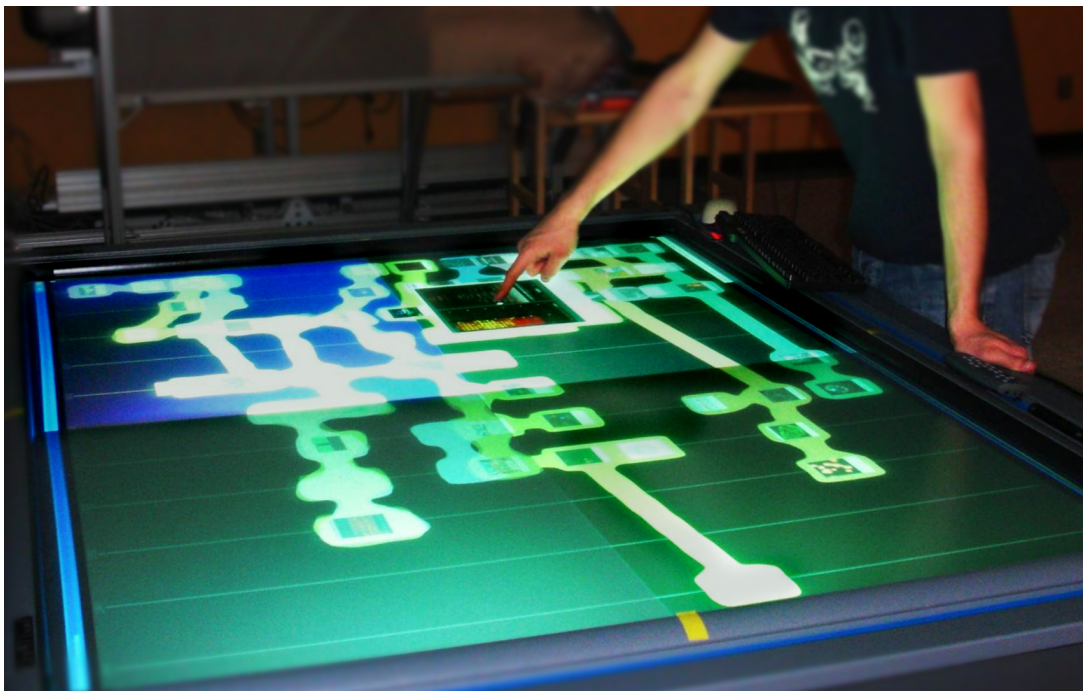


FIGURE 7.13: The research articles timeline Bubble Sets prototype can be explored using touch input on a large, high resolution digital table.

sketches (Tat, 2007), we provide an automatic creation of similar visualizations as a tool to support collaborative discussion and categorization of research articles (Figure 7.11).

In our prototype implementation, we use the data from Tat (2007): a collection of 60 research articles in the area of visualizing communication patterns. We manually collected characteristic images and abstracts from the electronic documents, but document thumbnails could easily be generated using the PDF icon generation method of Bauer *et al.* (2005). The layout used in this example provides hybrid spatial rights: the initial layout places article icons on a timeline according to the year of publication. The layout is then adjusted to improve density of the pre-determined set memberships using a force-directed layout algorithm, restricted to movement on the y-axis. Forces draw items that share at least one set toward one another and repel members of other sets. This layout mechanism provides for dense sets with minimal interference from other sets. However, in years where many articles appear on the timeline, interference cannot be avoided. In these cases, the surface routing algorithm creates an invisible structure of edges to route the set surfaces around obstacles.

Interactive movement of items along the y-axis allows for creation of customized views. Set membership is specified in the dataset, but can be changed at run time. To aid interactive classification of articles, the data can be queried for additional details, such as a larger image or the abstract, by clicking (desktop) or tapping (interactive tabletop) a document icon. The set boundary expands to contain the enlarged item, and a force-directed algorithm moves neighbouring items to maintain partial visibility (Figure 7.12). The interface is designed for use on a desktop computer, but can also be explored on a high-resolution touch-sensitive display, facilitating collaborative discussion (see Figure 7.13).

7.5.2 *Sets over Scatterplots*

Bubble Sets were developed to aid in machine translation research, but the technique is generally applicable to visualization problems outside the linguistic domain. Scatterplots have clearly defined spatiality due to the numerical positioning of items. We add Bubble Sets to a reimplementation of the well known *GapMinder Trendalyzer* (Rosling, 2009). This scatterplot shows fertility rate against life expectancy and is animated over time. Data points represent countries, sized by population. Colour (and set

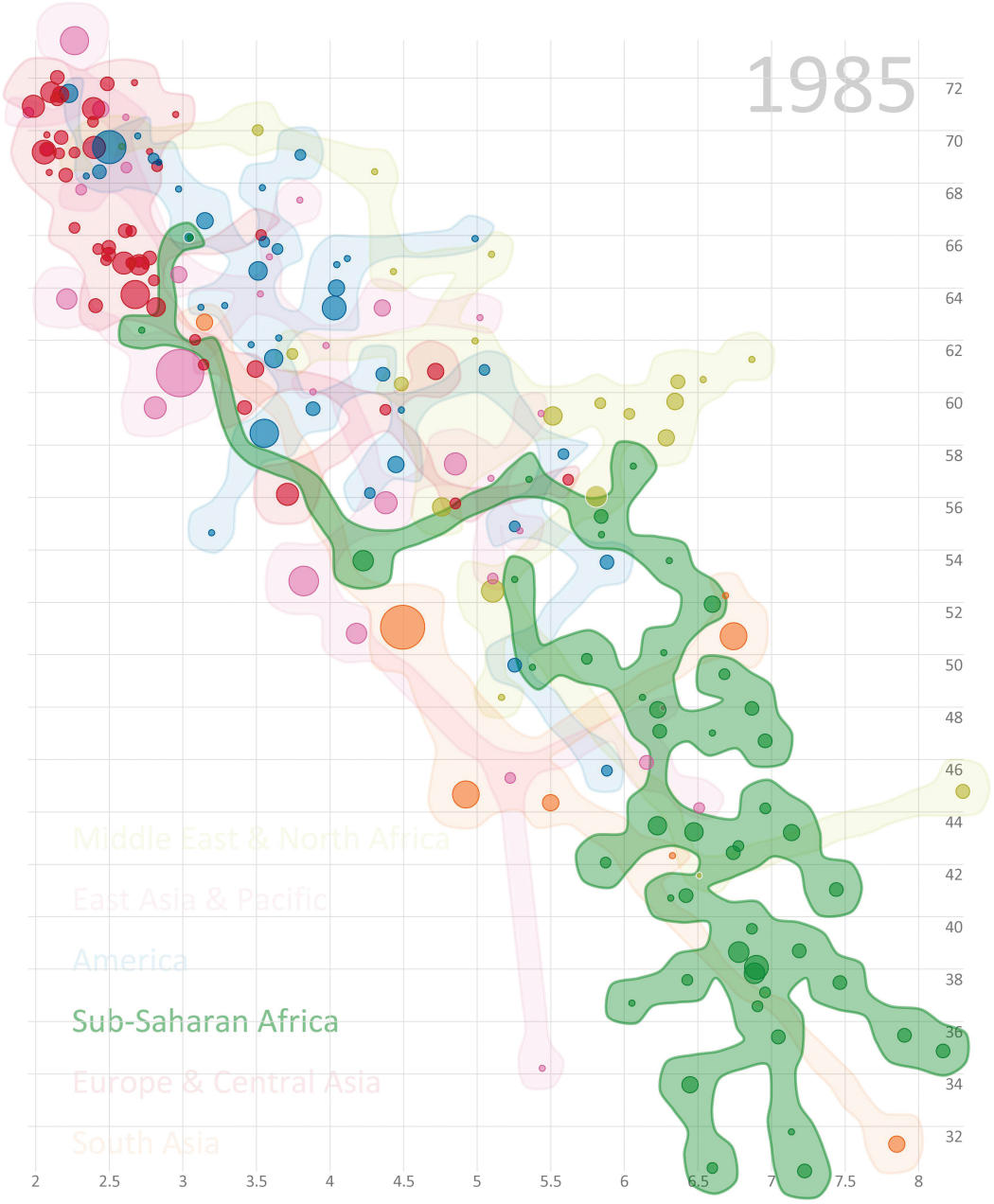


FIGURE 7.14: A scatterplot of fertility rate (x-axis) by life expectancy (y-axis) by country (circles, sized by population, coloured by continent). The plot can be animated over time. Hovering on a set member causes all non-members and other sets to be made transparent, clarifying set membership. Here, enclosure eases discovery of the outliers in the upper left, as well as giving a general impression about the spatial distribution of the set.

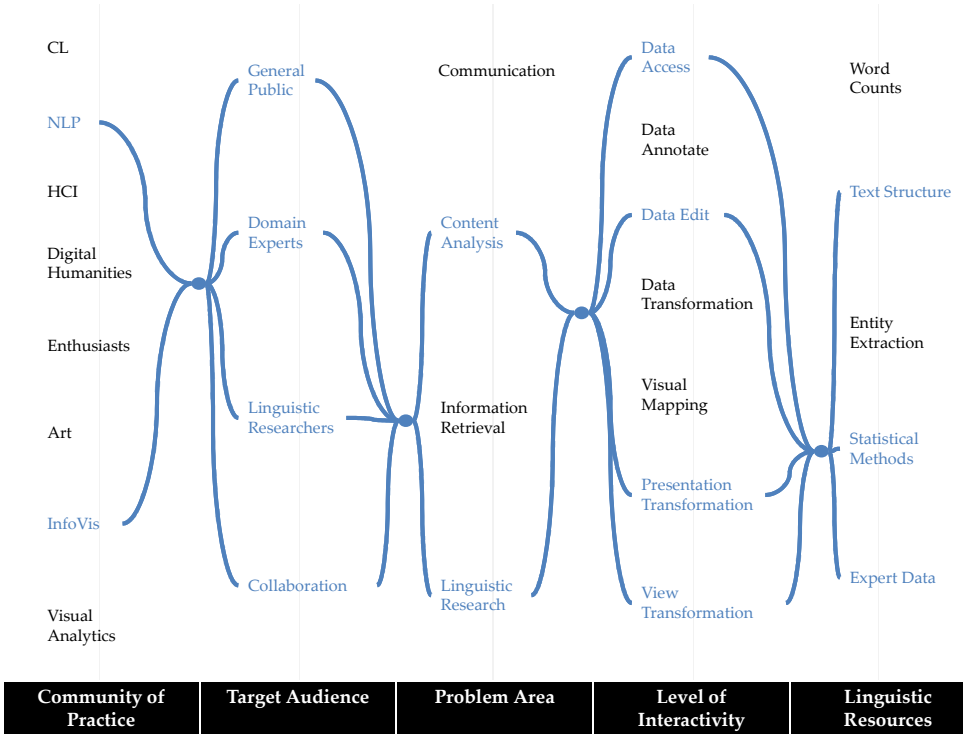


FIGURE 7.15: Bubble Sets in the space of linguistic visualization.

membership) is defined by the continent. The grouping of the sub-Saharan Africa countries, highlighted in Figure 7.14, reveals that while most of the countries in this set had high fertility rates and low life expectancies in 1985, there are two outliers, Mauritius and Reunion, which are islands in the Indian Ocean. As the data set includes data for many years, and since Bubble Sets are calculated at interactive rates, the temporal changes can be convincingly shown through animation.

7.6 SUMMARY

Guided by the needs identified through collaboration with MT researchers, we have contributed Bubble Sets: a method for automatically drawing set membership groups over existing visualizations with different degrees of requirements for primary spatial rights. In contrast to other overlaid containment set visualizations, Bubble Sets maximizes set membership

inclusion and minimizes inclusion of non-set members. In fact, Bubble Sets can guarantee that all set members will be within one container, as opposed to the more common multiple disjoint containers. While Bubble Sets cannot guarantee non-set member exclusion, the care taken with the routing algorithm minimizes these occurrences.

Within the *space of linguistic visualization*, the applicability of Bubble Sets is quite broad. Applications of the technique can be targeted at all of the target audience groups. Our examples include applications for content analysis of a document corpus, and linguistic research in MT. The level of interactivity will vary for each specific application. Generally, data access, data edit (assign sets), presentation transformations, and view transformations are provided. Adding data annotate to facilitate collaboration would be a trivial extension. The MT case study uses statistical models and expert data. The research articles timeline case study uses information about text structure and expert-annotated categorizations of articles. These dimensions are summarized in Figure 7.15.

8

REVEALING RELATIONS AMONGST VISUALIZATIONS

All these constructions and the laws connecting them can be arrived at by the principle of looking for the mathematically simplest concepts and the link between them.

— Albert Einstein, 1921

As information visualizations continue to play a more frequent role in information analysis, the complexity of the queries for which we would like visual explanations also continues to grow. While creating visualizations of multi-variate data is a familiar challenge, the visual portrayal of two sets of relationships, one primary and one secondary, within a given visualization is relatively new (*e.g.*, Fekete *et al.*, 2003; Holten, 2006; Neumann *et al.*, 2005). With VisLink, we extend this direction, making it possible to reveal relationships, patterns, and connections between two or more primary visualizations. VisLink enables reuse of the spatial visual variable, thus supporting efficient information encoding and providing for powerful visualization bridging which in turn allows inter-visualization queries.

For example, consider a linguistic question such as whether the formal hierarchical structure as expressed through the `IS-A` relationships in WordNet (Miller *et al.*, 2007) is reflected by actual semantic similarity from usage statistics. This is best answered by propagating relationships between two visualizations: one a hierarchical view of WordNet `IS-A` relationships and the other a node clustering graph of semantic similarity relationships. Patterns within the inter-visualization relationships will reveal the similarities and differences in the two views of lexical organization.

Demonstrating the *linguistic visualization divide*, the idea of viewing the patterns in edges between two 2D plots of related items has previously been expressed as a static information graphic in the computational linguistics (CL) community. For example, Figure 8.1 describes the concept of signal-to-meaning mappings to model language in a manually-created sketch (Brighton and Kirby, 2006). We do not know of any previous interactive systems demonstrating this concept.

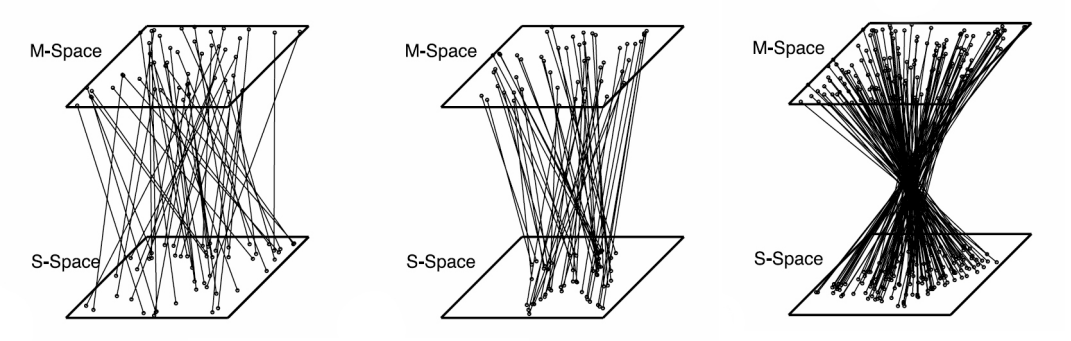


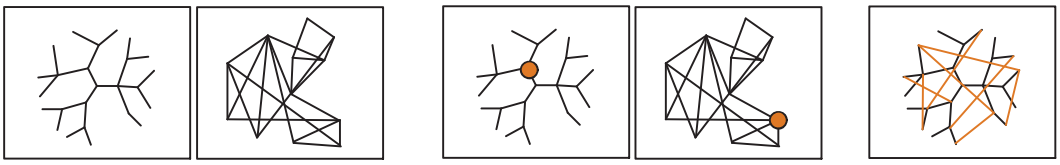
FIGURE 8.1: Brighton and Kirby (2006) illustrate the concept of degrees of compositionality in language. At left, signals and meanings are randomly mapped (representing a non-compositional language). In the center, a language where similar meanings do map to similar signals. At right is a representation of a fully distance-correlated mapping (compositional). *Reprinted with permission.*

In this chapter, we will describe VisLink, a new visualization technique which bridges this divide by realizing the display and querying of multiple 2D visualizations in 3D space, each with its own use of spatial organization and each placed on its own interactive plane. These planes can be positioned and re-positioned supporting inter-visualization comparisons; however, it is VisLink's capability for supporting cross-representational queries that is our main contribution. Propagating edges between visualizations can reveal patterns by taking advantage of the spatial structure of both visualizations. In the next few pages on we will explain our new visualization technique in comparison to existing multi-relationship visualizations.

8.1 FORMALIZING VISUALIZATIONS OF MULTIPLE RELATIONS

VisLink extends existing approaches to visualizing multiple relationships by revealing relationships amongst visualizations while maintaining the *spatial rights* of each individual relationship type. In order to discuss more precisely the distinctions between previous work and our contribution, we will first introduce some notation for describing multiple view visualizations.

Given a data set, D_A, and a set of relationships, R_A, on D_A, we will write this as $R_A(D_A)$. Note that with the relation R_A we are not referring to a strict mathematical function, but rather any relation upon a data set, for example, a type of edge among nodes in a general graph. A second set



(a) (b) (c)

FIGURE 8.2: Current approaches to comparing visualizations include (a) manual comparison (printed diagrams or separate programs), (b) coordinated multiple views (linked views with highlighting), and (c) compound graphs (layout based on one relationship, other relationships drawn upon it).

of relationships on the same data set would be $R_B(D_A)$, while the same set of relationships on a different but parallel data set would be $R_A(D_B)$. For example, if the data set D_A was housing information in Montreal, an example of R_A could be the specific house to property tax relation $R_A(D_A)$ and a different relationship R_B could be the house size as related to the distance from transit routes $R_B(D_A)$. Then an example $R_A(D_B)$ would be property tax on houses in Toronto. Creating a first visualization, Vis_A, of these relationships $R_A(D_A)$ we will write $Vis_A \rightarrow R_A(D_A)$ (for example, a geographic map with houses coloured based on their property tax). A second visualization, Vis_B, of the same set of relationships would be $Vis_B \rightarrow R_A(D_A)$ (for example, a histogram of number of houses in each property tax range).

In the remainder of this section, we use this notation to define, compare, and contrast each of the current approaches to relating visualizations. We will show how VisLink provides capability beyond what is currently available.

8.1.1 *Individual Visualizations*

As a viewer of any given set of visualizations it is possible to do the cognitive work of developing cross visualization comparisons. For instance, visualizations can be printed and one can, by hand with pen and pencil, create annotations and/or new visualizations to develop the comparisons needed for the current task. Any relations on any data may be compared manually in this way (see Figure 8.2a).

8.1.2 *Coordinated Multiple Views*

Coordinated views provide several usually juxtaposed or tiled views of visualizations that are designed to be of use in relationship to each other (*e.g.*, Snap-Together Visualization (North and Shneiderman, 2000)). These can be of various flavours such as Vis_A, Vis_B and Vis_C of $R_A(D_A)$ or perhaps Vis_A of $R_A(D_A)$, $R_B(D_A)$ and $R_C(D_A)$. The important factor for this visualization comparison discussion is that these coordinated views can be algorithmically linked such that actions and highlights in one view can be reflected on other views. Coordinated views allow for reuse of the spatial visual variable, thus each relationship type is afforded spatial rights. The temporarily activated visual connections can be a great advantage over finding the related data items manually but the relationships themselves are not explicitly visualized (see Figure 8.2b).

8.1.3 *Compound Graph Visualizations*

There are now a few examples of compound graph visualizations, such as overlays on *Treemaps* (Fekete *et al.*, 2003), *ArcTrees* (Neumann *et al.*, 2005), and *Hierarchical Edge Bundles* (Holten, 2006). Figure 8.2c shows a simple diagram of this. Compound graph visualizations are created as follows:

GIVEN: Data set D_A, containing two (or more) types of relationship: $R_A(D_A)$, $R_B(D_A)$, ..., $R_N(D_A)$.

PROBLEM: Show multiple relationship types on the same visualization.

STEP 1: Choose a relationship type, *e.g.*, R_A, to be the primary relationship.

STEP 2: Create a visualization $Vis_A \rightarrow R_A(D_A)$, providing an appropriate spatial layout. Since spatial organization is such a powerful factor in comprehending the given relationships, we refer to this as giving R_A *spatial rights*.

STEP 3: Create a visualization of $R_B(D_A)$ (and any other desired secondary relations) atop $Vis_A \rightarrow R_A(D_A)$.

This in effect creates $Vis_A \rightarrow R_A, R_B(D_A)$ using the spatial organization of $Vis_A \rightarrow R_A(D_A)$. While this is an exciting step forward in comparative visualization, note that $R_B(D_A)$ has no spatial rights of its own. That is,

while viewing how the relationships in $R_B(D_A)$ relate to $R_A(D_A)$ is possible, there is no access to a visualization $Vis_B \rightarrow R_B(D_A)$. Hierarchical Edge Bundles (Holten, 2006) started an interesting exploration into using the spatial organization of $R_A(D_A)$ to affect the readability of the drawing of $R_B(D_A)$ atop $Vis_A \rightarrow R_A(D_A)$ and also indicated possibilities of addressing the readability needs of $R_B(D_A)$ by altering the spatial drawing of $Vis_A \rightarrow R_A(D_A)$ so that $R_B(D_A)$ and $R_A(D_A)$ occupy different spatial areas. This gives $R_B(D_A)$ partial spatial rights in that its presence affects the $Vis_A \rightarrow R_A(D_A)$ layout.

8.1.4 *Semantic Substrates Visualizations*

Shneiderman and Aris (2006) introduce Semantic Substrates, a visualization that is both quite different and quite similar in concept to VisLink. We will use our notation to help specify this:

GIVEN: Data set D_A and a set of primary relationships $R_A(D_A)$.

PROBLEM: A given unified visualization creates too complex a graph for reasonable reading of the visualization.

STEP 1: Partition the data set D_A into semantically interesting subsets, $D_{A_1}, D_{A_2}, \ldots, D_{A_n}$.

STEP 2: Use the same visualization Vis_A, with spatial rights, to create visualizations of the subsets $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2})$, $\ldots$, $Vis_A \rightarrow R_A(D_{A_n})$.

STEP 3: Juxtapose one or more of $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2})$, $\ldots$, $Vis_A \rightarrow R_A(D_{A_n})$, aligned in a plane.

STEP 4: Draw edges of $R_A(D_A)$ across $Vis_A \rightarrow R_A(D_{A_1})$, $Vis_A \rightarrow R_A(D_{A_2})$, $\ldots$, $Vis_A \rightarrow R_A(D_{A_n})$ to create $Vis_A \rightarrow R_A(D_A)$.

8.1.5 *VisLink Visualizations*

Now we will use our notation to clarify the contribution of the VisLink visualization:

GIVEN: Data set D_A and a set of primary relationships $R_A(D_A)$, $R_B(D_A)$, $\ldots$, $R_N(D_A)$.

PROBLEM: Provide a visualization that aids in improving the understanding of $R_A(D_A)$, $R_B(D_A)$, ..., $R_N(D_A)$ by indicating how one set of relationships is related to the structure in another.

STEP 1: Create visualizations $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_A)$, ..., $Vis_N \rightarrow R_N(D_A)$, each with full spatial rights for any of $R_A(D_A)$, $R_B(D_A)$, ..., $R_N(D_A)$ that are of interest.

STEP 2: Place selected visualizations $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_A)$, ..., $Vis_N \rightarrow R_N(D_A)$ on individual planes to support varying types of juxtaposition between visualizations (at this point we are limiting these to 2D representations).

STEP 3: Draw edges of second order relations $T(R_A, R_B, \ldots, R_N(D_A))$, from $Vis_i \rightarrow R_i(D_A)$ to $Vis_{(i+1)} \rightarrow R_{(i+1)}(D_A)$ and $Vis_{(i-1)} \rightarrow R_{(i-1)}(D_A)$ to create VisLink inter-plane edges between neighbouring planes.

So, where Semantic Substrates operates with a single visualization type and single relation across multiple subsets of a data set, VisLink can operate on multiple visualization types and multiple relationship types on a single dataset. A natural extension of VisLink is to inferred or indirect relations across multiple data sets:

GIVEN: Data sets D_A, D_B, ..., D_N and the existence meaningful relationships, $T(D_i, D_j)$, among datasets such that (i, j) are any of A, B, ..., N.

VISUALIZE: VisLink can be used with no further extensions to relate $Vis_A \rightarrow R_A(D_A)$, $Vis_B \rightarrow R_B(D_B)$, ..., $Vis_N \rightarrow R_N(D_N)$, by using $T(D_i, D_j)$ to create inter-plane edges. An example of cross-dataset visualization is presented in Section 8.4.

We have presented a series of multi-relation visualizations, differing in the level of visual and algorithmic integration between relations and the amount of spatial rights accorded to secondary relations. VisLink can be used equivalently to any of the mentioned multi-relation visualization approaches (see Figure 8.3a–c) and extends the series to simultaneously provide equal spatial rights to all relations for which a visualization can be created, along with close visual and algorithmic integration of different relations (see Figure 8.3d).

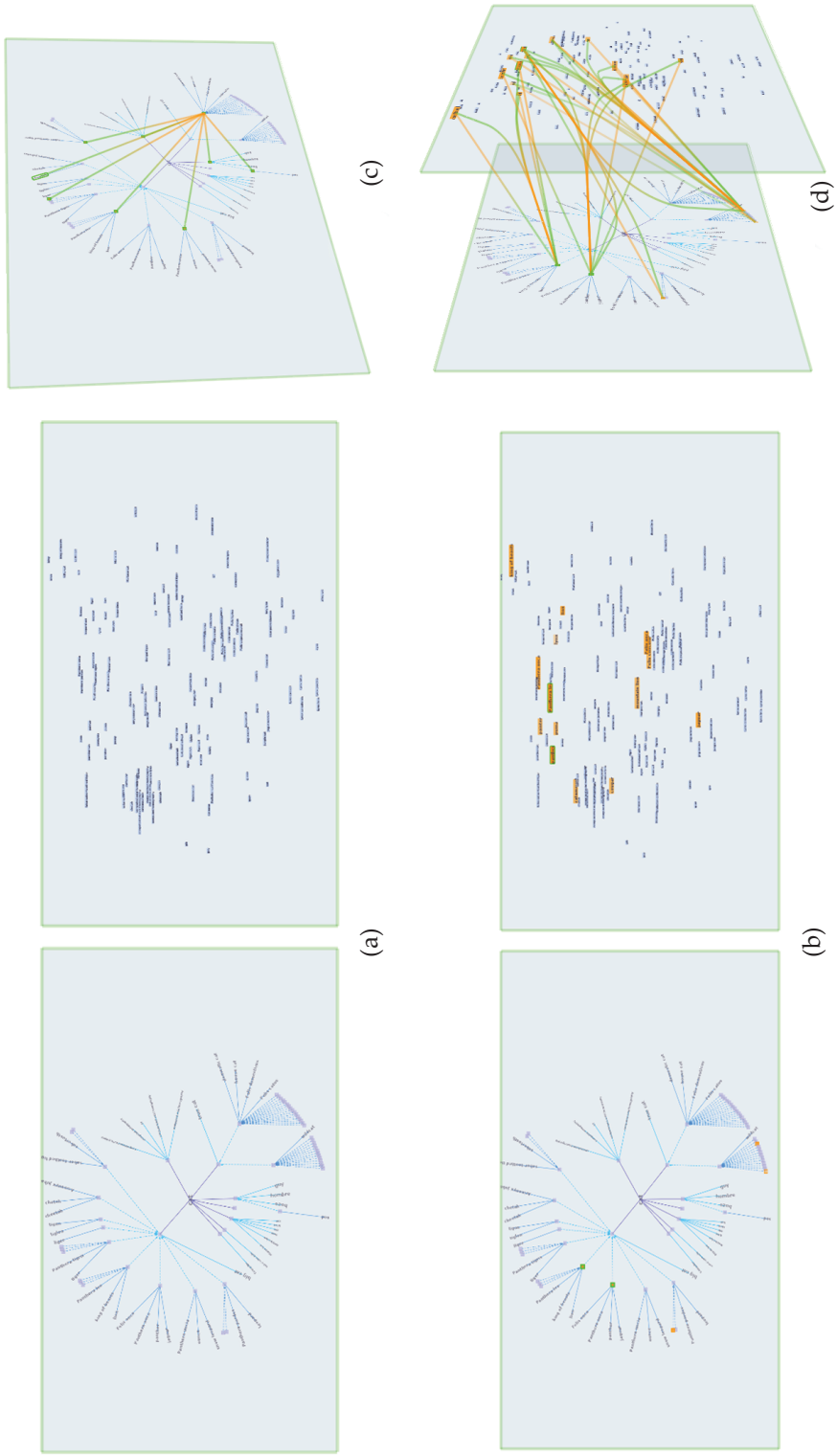


FIGURE 8.3: VisLink encompasses existing multiple views techniques of (a) manual comparison, (b) coordinated multiple views, and (c) compound graphs. VisLink extends this continuum to direct linking of any number of multiple views (d).

8.2 DESIGN OF VISLINK

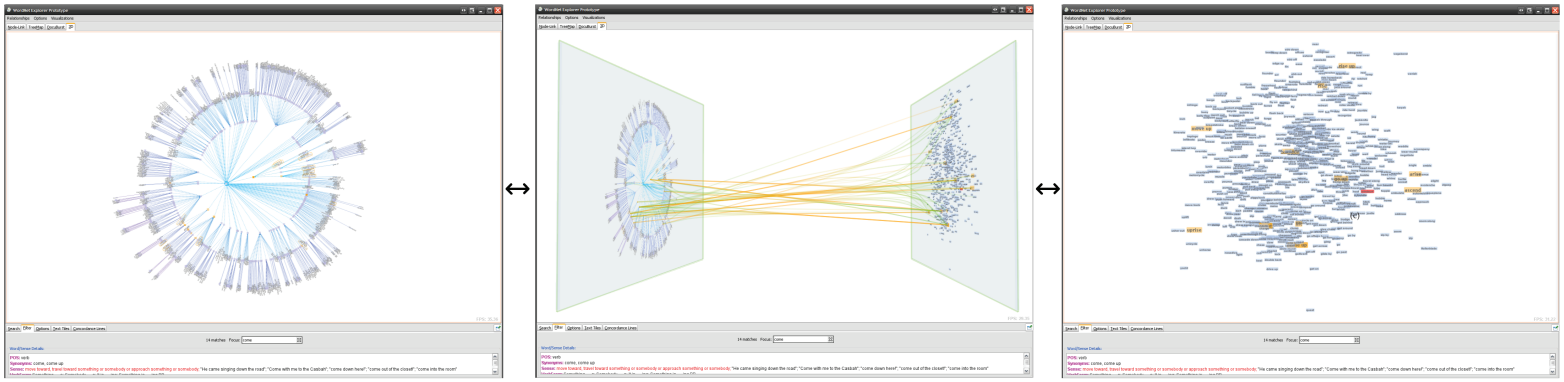
In order to provide for a visualization space in which multiple data-related visualizations can be analyzed, we have developed VisLink. We start our explanation with a very brief description of the lexical data set and the lexical data relationships which are used to illustrate VisLink's functionality and interactive capabilities. Next we show a sample set of 2D lexical visualizations displayed on visualization planes within VisLink, followed by the possible interactions with these visualization planes. Then the inter-visualization edges are explained and the ability to use inter-plane edge propagation to answer complex queries is presented.

8.2.1 *Visualizations of Lexical Data*

The example figures in this chapter are drawn from application of VisLink to a lexical data set. This is an area of interest to computational linguists, and several visualizations using lexical data have been reported including DocuBurst (Chapter 5) and several others (*e.g.*, Barrière and St-Jacques, 2005; Dunn, 2007; Kamps and Marx, 2002a; Manning *et al.*, 2001; Ploux and Ji, 2003).

Using our formalism, we have a dataset D_A containing all the words in the English language. There are many types of relationships among words, for example, the lexical database WordNet (Miller *et al.*, 2007) describes the hierarchical `IS-A` relation over synsets, which are sets of synonymous words. For example, *{lawyer, attorney}* `IS-A` *{occupation, job}*. The `IS-A` relation is also called hyponymy, so *chair* is a hyponym of *furniture*. We use hyponymy to build animated radial graphs (Yee *et al.*, 2001), which serve as our $Vis_A \rightarrow R_A(D_A)$. Synsets are shown in the radial graph as small squares, and the synonymous words that make up the set are shown as attached, labelled, nodes. An example 2D radial hyponymy graph is in Figure 8.4a.

Words can also be related by their similarity. Similarity can be a surface feature, for example, orthographic (alphabetic) similarity, or it can be based on underlying semantics. We use a force-directed layout to perform similarity clustering on words (Battista *et al.*, 1999). In our examples we use orthographic similarity, so that all words are connected to all others by springs whose tension coefficient is inversely related to number of consecutive character matches in the substring, starting at the beginning.



(a) (b) (c)

FIGURE 8.4: Viewing modes. (a) 2D equivalency view of plane one, showing hyponyms of verb 'move', with highlighted search results for 'come'. (b) Search results on plane one activate inter-plane edges, visible in 3D mode. Nodes connected to search results are highlighted on plane two, a similarity clustering of words related to 'move'. Propagated results are also visible when plane two is viewed in 2D equivalency mode (c).

Words that start with the same letters will cluster together. This is a very different structure than hyponymy and serves as $Vis_B \rightarrow R_B(D_A)$. An example 2D alphabetic clustering visualization is in Figure 8.4c. We have also experimented with clustering using the semantic similarity measures implemented by Pedersen *et al.* (2005), for example similarity as measured by lexical overlap in the dictionary definitions of words. However, those measures did not produce visible clusters and further investigation is needed into the appropriate relationship between the similarity measure and the spring coefficient.

Using VisLink, we investigate relations between the hyponymy layout of synsets and the orthographic clustering layout of words. With this, we can investigate questions such as: do some synsets contain high concentrations of orthographically similar words?

Data is loaded into the VisLink lexical visualization by looking up a synset in WordNet to root the hyponymy tree. The orthographic clustering is then populated with the relevant words from the dataset.

8.2.2 *Navigation and Plane Interaction*

VisLink is a 3D space within which any number of 2D semi-transparent visualization planes are positioned. These visualization planes act as virtual displays, upon which any data visualization can be drawn and manipulated. They can be rotated and shown side by side similar to multi-program or coordinated views, or rotated in opposition with included connections. Interaction and representation with each plane remains unchanged (representations do not relinquish any *spatial rights* nor any *interaction rights*).

While VisLink is a 3D space, the visualization planes are 2D equivalents of a display, similar to windows in *Miramar* (Light and Miller, 2002) or view-ports in the *Web Forager* (Card *et al.*, 1996). We provide view animation shortcuts to transition between 2D and 3D views. Similar to interaction provided by *Miramar*, any visualization plane may be selected, activating an animated transition in which the selected plane flies forward and reorients to fill display space. When a plane is selected, 3D interaction widgets and inter-plane edges are deactivated, and the display becomes equivalent to 2D (see Figure 8.4). Because VisLink visualization planes have the same virtual dimensions as the on-screen view-port, transition between 2D plane view and 3D VisLink view does not require any resizing

of the selected plane. When the plane is deselected, it falls back into the VisLink space, reverting to the original 3D view.

Interaction with the visualization on a visualization plane is always equivalent to 2D: mouse events are transformed to plane-relative coordinates and passed to the relevant visualization (irrespective of the current position and orientation of the plane). Visualizations can be manipulated directly in the 3D space (using equivalent-to-2D mode is not necessary). Thus interaction techniques developed for 2D visualizations become immediately available in VisLink. For example, we provide for a radial node-link view of the WordNet hyponymy (`IS-A`) relation, restricted with a generalized fish eye view to show only nodes of distance N or less from the central focus. The focus node can be reselected by a mouse click, activating radial layout animation (Yee *et al.*, 2001). Double clicking any node restricts the view to the tree rooted at that node, providing for drill-down capability. Drill down and other data reload interactions are propagated to all planes. Interaction techniques such as panning and zooming in 2D are provided by clicking and dragging on a visualization plane the same as one would on an equivalent stand-alone 2D visualization.

In addition to interaction with the visualizations on VisLink planes, we also provide for interaction with the planes themselves. While the usual capabilities for navigation in a 3D space (pan, zoom, rotate of camera position) are available in VisLink, in providing a 3D perspective projection virtual space, we must address the difficulties that arise from 6-degrees-of-freedom (DOF) control with 2-DOF input devices (Bowman *et al.*, 2005). Free navigation can result in disorientation and non-optimal viewing positions, while free manipulation of 3D objects can result in difficulty achieving precise desired positioning.

Therefore, we also provide shortcuts for cinematic animated repositioning of the camera and planes to preset viewpoints (Light and Miller, 2002). These viewpoints allow visualization planes to be viewed from the front (planes parallel and side by side) (see Figure 8.5a), with relative plane orientation of book view (planes perpendicular and meet at an edge) (see Figure 8.5b), top (see Figure 8.5c and d), or in opposition (planes parallel and stacked) (see Figure 8.5d and e). By choosing one of these viewpoints, users can recover from any disorienting manipulation.

As a solution to 2D plane interaction in a 3D space, we follow McGuffin *et al.* (2003) and provide for manipulation of visualization plane position and orientation using a set of restricted movement widgets. Edge widgets

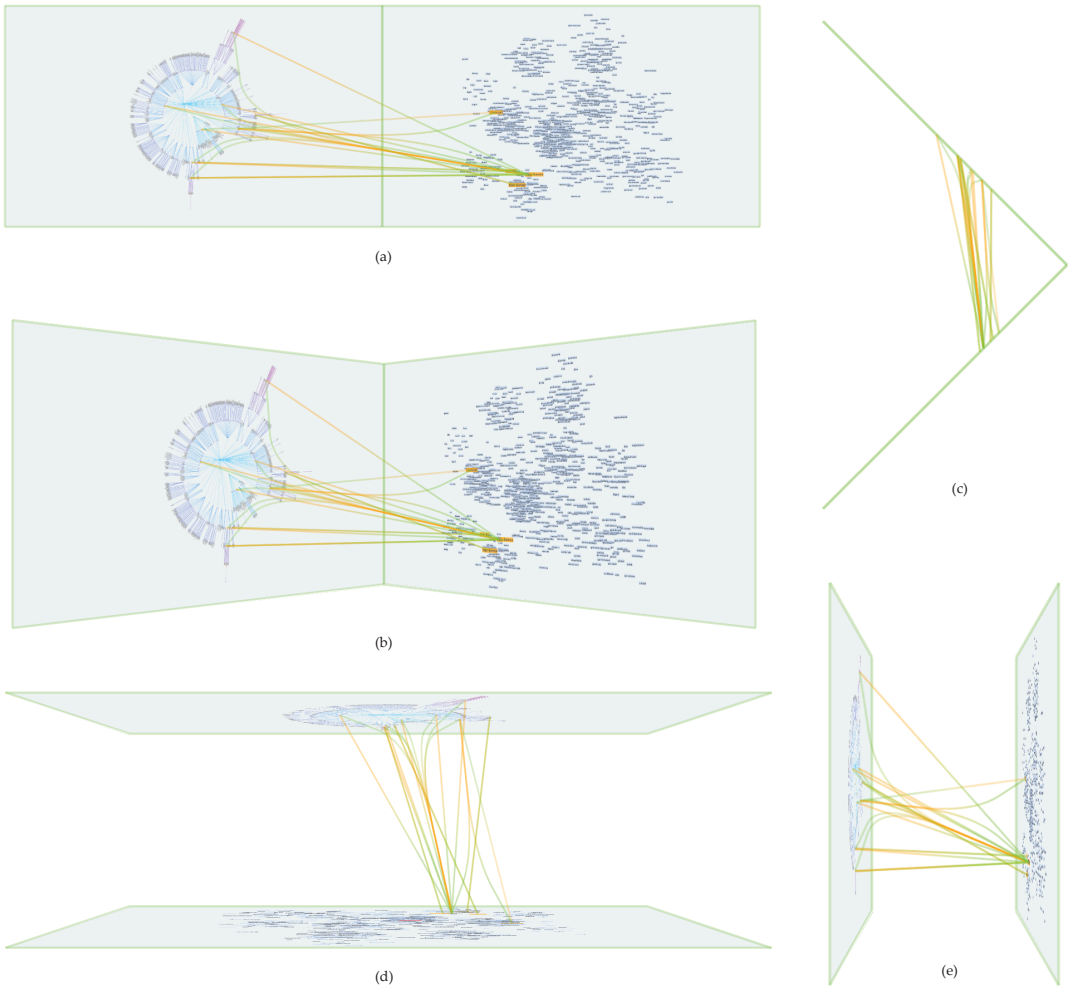
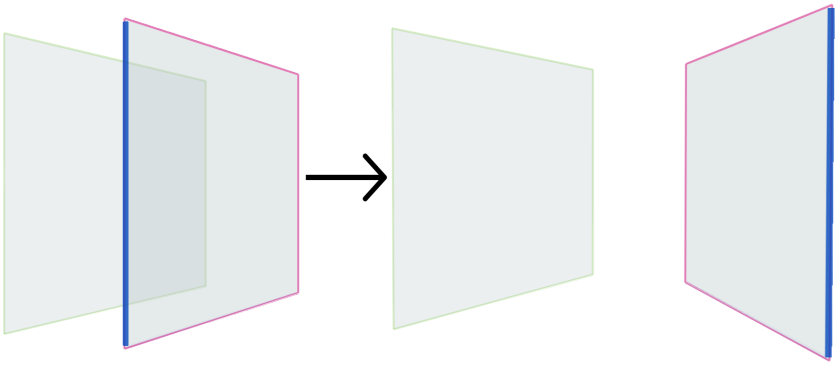
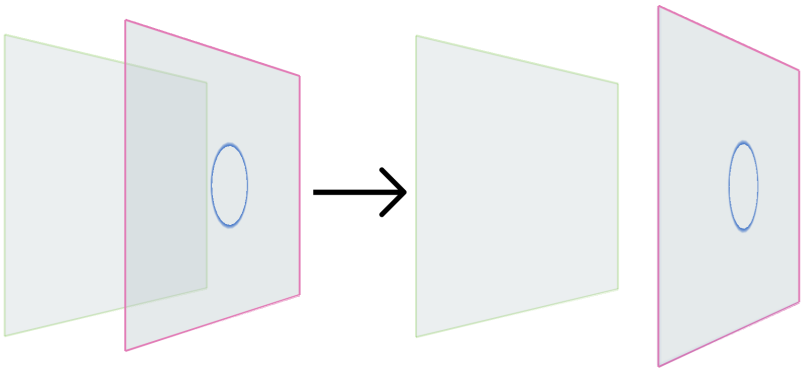


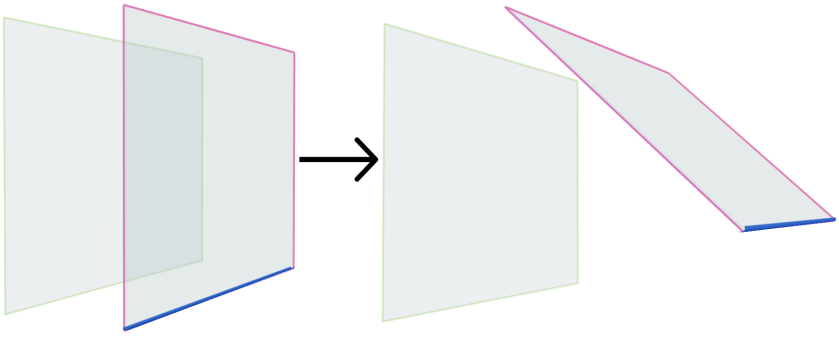
FIGURE 8.5: Keyboard shortcuts provide for animated transition to default views, easing navigation in the 3D space. Views are (a) flat, (b) book, (c) book top, (d) top, and (e) side.



(a) Side *book pages* rotation widget.



(b) Center *accordion* translation widget.



(c) Bottom *garage door* rotation widget.

FIGURE 8.6: Visualization planes are independently manipulated with three types of widgets (blue areas).

provide for hinge movement (up to 90 degrees) about the opposite edge, and a center widget provides for translation, accordion style, along the axis between the planes (see Figure 8.6). Widgets become visible when the pointer is over their position, otherwise they are hidden from view to prevent data occlusion.

8.2.3 *Adding Inter-Plane Edges*

Edges are drawn in 3D to bridge adjacent visualization planes. Relationships between the visualizations can either be direct (nodes representing the same data are connected across planes) or indirect (items on different planes have relations defined within the data).

For example, in our lexical visualization, we examine the formal structure of WordNet hyponymy (the `IS-A` relation) on one plane, and the clustering of words based on their similarity on another. The inter-plane relationship in this case is direct: nodes on plane one represent the same data as nodes on plane two. In this case, it is the difference in the spatial organization of the layouts that is of interest. In essence, the pattern of inter-plane edges reveals a second-order relation: the relationship between different types of node relations on the same data. If the clustering by similarity approximates the formal structure, edges from synonyms in the structured data will go to the same cluster (*i.e.*, edges from synonyms will be parallel).

Indirect relations can also be visualized. For example, a visualization plane could be populated with a general graph about self-declared friendships in a social networking system. A second visualization plane could be populated with a tag cloud from a folksonomy, for example a bookmark sharing database. A third visualization plane could be populated with a visualization of the hypertext links between bookmarked pages. The three types of indirect inter-plane connections could be derived from three cross-dataset rules: `Person used Tag`, `Page tagged with Tag`, and `Person bookmarked Page`. With effective inter-plane edge management and data filtering, patterns between planes in such a visualization could reveal people who share tagging habits, or bookmarked pages with similar tag sets.

All inter-plane edges are specified with a single source node on plane i and one or more target nodes on plane j. Single source to single target edges are drawn as straight lines. Single source to many target edges

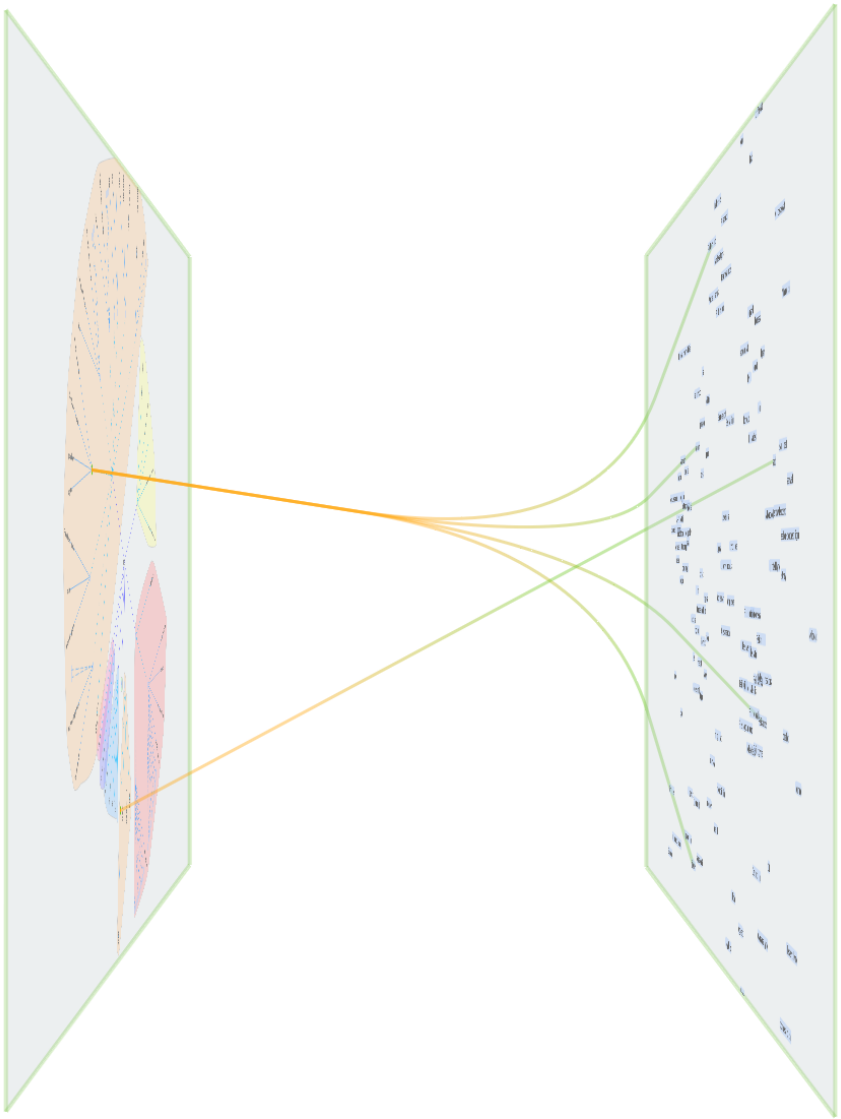


FIGURE 8.7: VisLink inter-plane edge detail: one-to-one edges are straight, one-to-many edges are bundled. Alpha blending provides for stronger appearance of bundled edges.

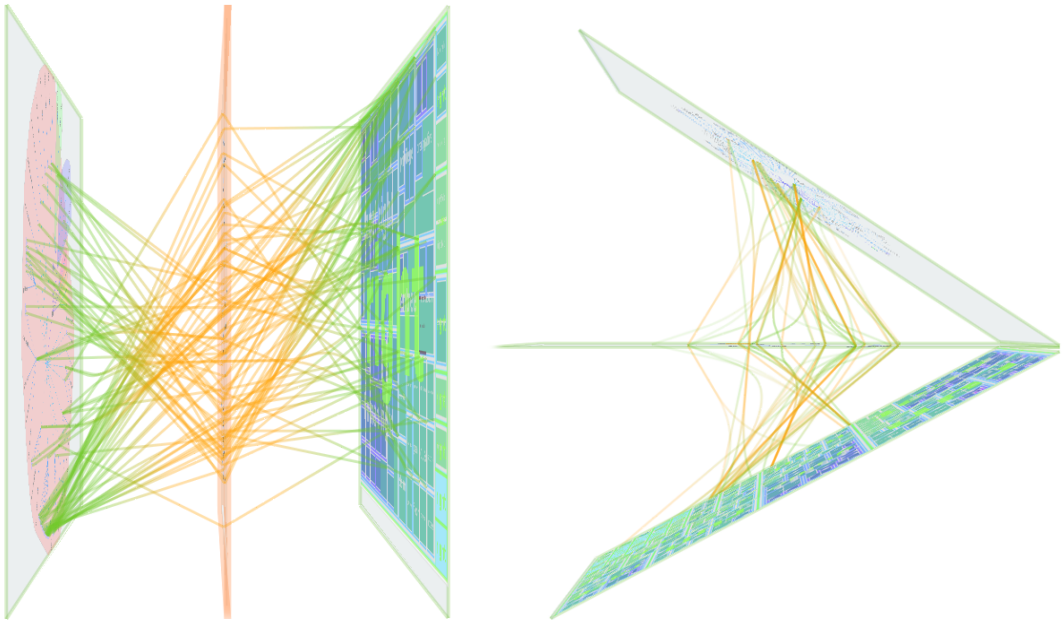
are drawn using multiple curves calculated with corner-cutting (Chaikin, 1974). For each curve from the source to a target, the starting control point is set as the source node, a middle control point is set as the average (world coordinates) position of all target nodes and the source, and the end point is set as the target. Five iterations of corner-cutting provide for smooth curves which start along the same straight line and then diverge as they approach their targets. By using alpha blending, the more semi-transparent curves that are coincident, the stronger the bundled edges appear (see Figure 8.7). Inter-plane edge positions are recalculated as appropriate so that edges remain fluidly attached to their source and target nodes throughout all manipulations of the constituent visualizations, plane positions, and the 3D viewpoint.

For visual clarity, edges are drawn between items on adjacent planes only. For more than two visualization planes, if the data contains relations among all visualizations, these relations can be explored by reordering the visualization planes using the center translation (accordion) widget to move planes along the inter-plane axis. As a plane passes through another, the rendering is updated to show the relations between the new neighbours. Similar to axis ordering in parallel coordinates plots (Inselberg and Dimsdale, 1990), the ordering of visualization planes strongly effects the visibility of interesting patterns in the data. Investigation into methods for choosing plane orderings is left for future research.

8.2.4 *Using Inter-Plane Edges*

Inter-plane edges can be revealed either on a per-plane basis (see Figure 8.8) or a per-node basis (see Figure 8.9). Activating an entire plane can reveal structural patterns that may exist between the visualizations, while individual node activation provides for detailed views of particular relations.

We provide for spreading node activation between planes, which adds additional analytic power to VisLink. When a node is manually activated on one plane, it is highlighted in orange with a green border and all inter-plane edges originating at that node are revealed. The target nodes for those edges are then activated. Edges originating at these nodes are then drawn and the activation is propagated iteratively up to a user-selected number of *reflections* between planes. Deactivation of a node reverses



(a) (b)

FIGURE 8.8: Three visualization planes. (a) Side view with center plane (lexical similarity clustering) activated, indicated by its orange frame. (b) Top view in *book* orientation. Several nodes are activated on the outer planes, spreading their activation through inter-plane edges.

the process, spreading the deactivation and hiding edges. The level of activation exponentially decays with each iteration.

Nodes are assigned activation values from 0 (deactivated) to 1 (manually activated by user through selection, search, or plane activation). Node activation values determine inter-plane edge visibility: edges between nodes with non-zero activation are revealed. Level of activation is inversely related to the alpha transparency of activated nodes and the inter-plane edges. So, the more transparent an activated node or edge, the further it is from a user-selected fully-activated node. Edge colour is used to indicate the direction of spreading activation. For each edge, the third closest to the source of edge activation is orange, the middle third is interpolated from orange to green and the final third, closest to the edge target, is green. Along with edge transparency decay, edge colouration will help an analyst follow the path of spreading activation. However, tracing a series of edges across planes may be a difficult task, even with the visual support provided through colouration and transparency. We plan to investigate

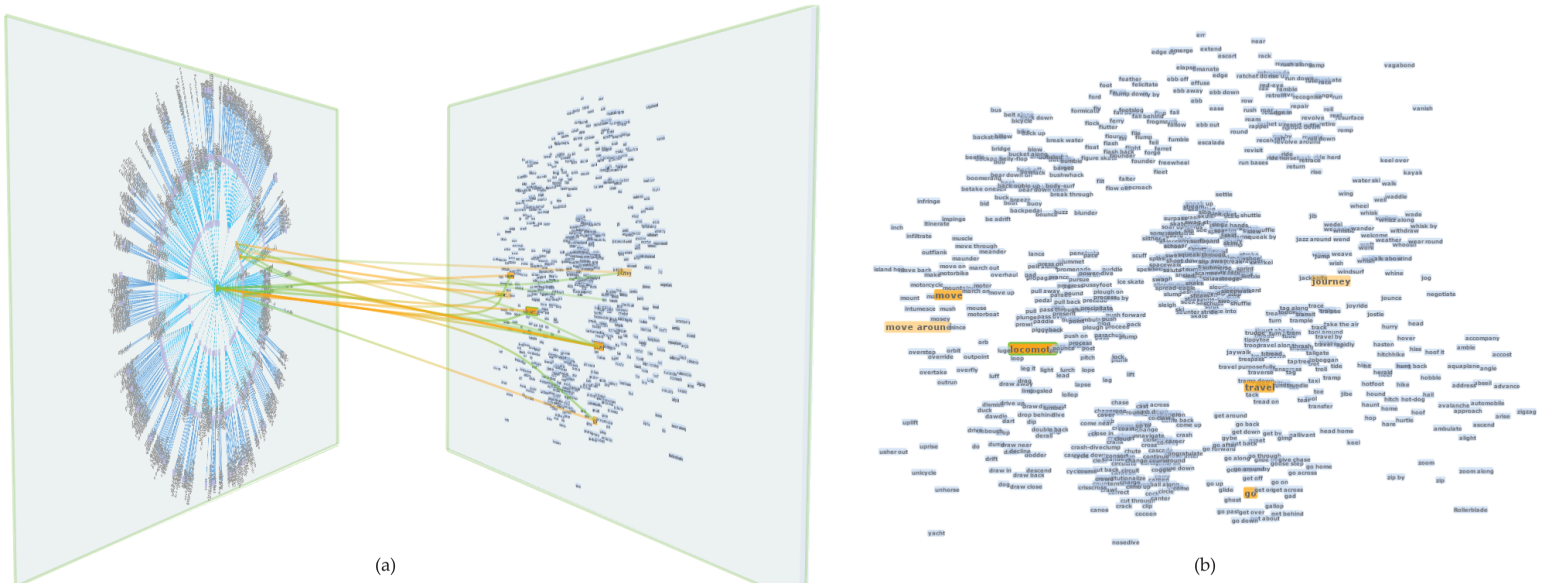


FIGURE 8.9: (a) Node activation and edge propagation. (b) Nodes highlighted through spreading activation (orange without green border) reveal the alphabetic clustering of synonyms of the manually activated node ('locomotion', orange with green border), as discovered through spreading activation to the WordNet hyponymy graph.

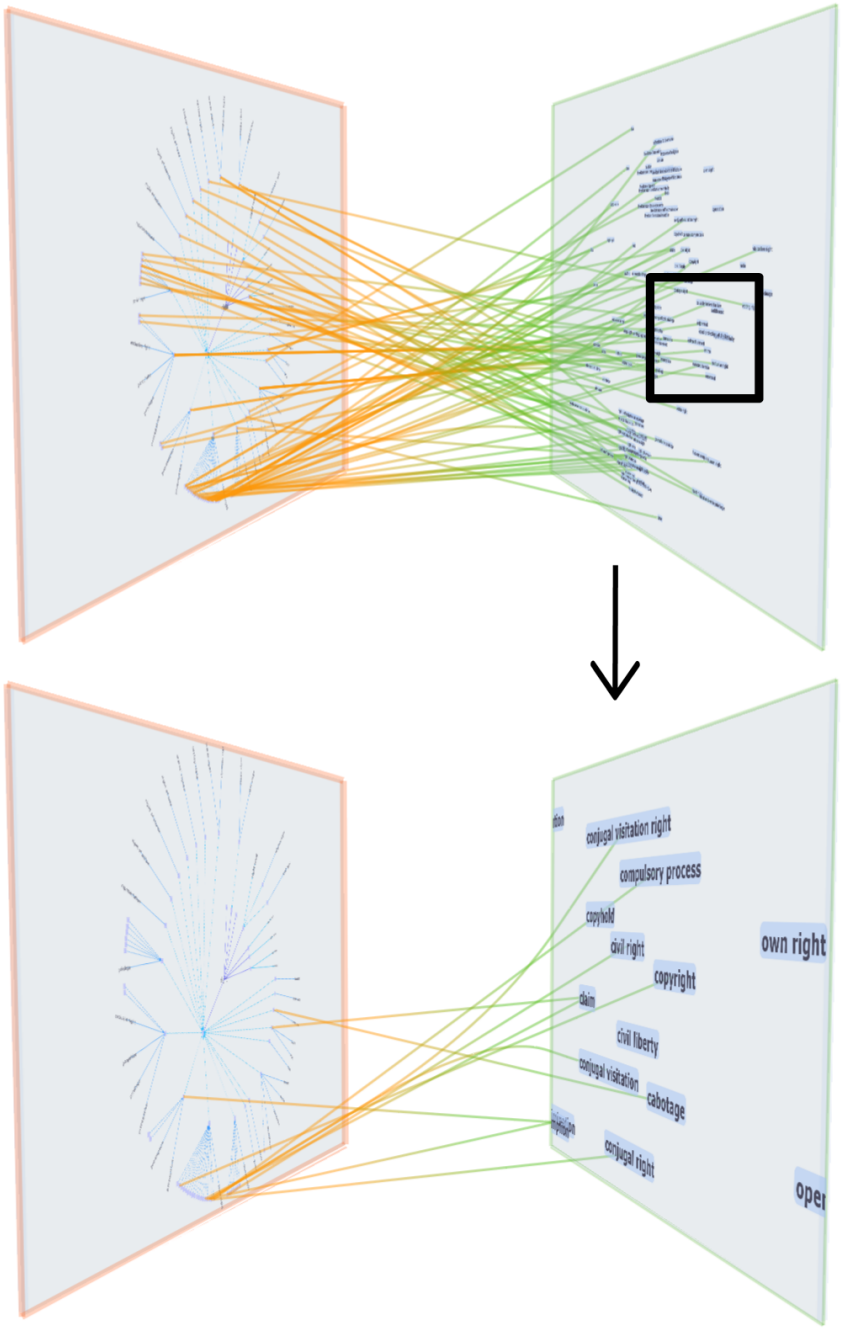


FIGURE 8.10: The left plane is activated, revealing all edges from it. Through a click and drag on the right plane, a 2D zoom is performed, isolating a cluster of interest. The inter-plane edges are filtered in real time to show only those connecting visible nodes, revealing that this lexical cluster is related to a region of the WordNet hyponymy tree near the bottom.

techniques such as animated edge propagation to help trace relationships amongst visualizations.

Inter-plane edges support cross-visualization queries. For example, alphabetic clustering, while a common organization for word search, is not useful for finding synonyms. Using VisLink to propagate an edge from a selected word in the clustered graph to a WordNet hierarchy will find this word within its synset structure, propagating back will find its synonyms within their alphabetic structure, allowing quick answers to questions such as, "*Across all senses, which synonyms of 'locomotion' start with 't'?*" This analysis is illustrated in Figure 8.9.

Inter-plane edges are only shown among visible nodes. So, if a technique such as filtering through degree-of-interest or distance measures, or clipping through zooming and panning the visualization on a plane causes some nodes to be invisible, their edges are not drawn. This can be used as an advantage for exploring the space of inter-plane edges: by filtering the view on a plane, the inter-plane edges can also be filtered (see Figure 8.10). Conversely, search techniques can be provided to reveal and activate nodes that match a query, thereby also activating their inter-plane edges (recall Figure 8.4).

8.3 IMPLEMENTATION DETAILS

VisLink is implemented in Java, using the Java2D-Java Opengl (JOGL) bridge to import any Java2D rendering onto a visualization plane. We have augmented the popular *prefuse* interactive visualization toolkit (Heer *et al.*, 2005) with the `VisualizationPlane` class, which implements the same API as the default 2D prefuse `Display`, the `InterPlaneEdge` class, which handles edge drawing between planes, and a controller which places visualizations in 3D space and connects them with rendered 3D edges. The result is that our visualization plane can accept any prefuse visualization without any changes. Interaction techniques on prefuse visualizations are also handled equivalently. In addition to providing for easy integration of existing visualizations with VisLink, this implementation provides for efficient rendering of the 3D space, achieving frame rates greater than 45fps on standard hardware (Intel Core 2 Duo, 2.4GHz processor with an NVIDIA GeForce 8800 graphics card). Inter-plane edges can be specified in the data set by referencing source and target visualization plane and node indices, or can be defined by a rule, such as, "*Create inter-plane edges*

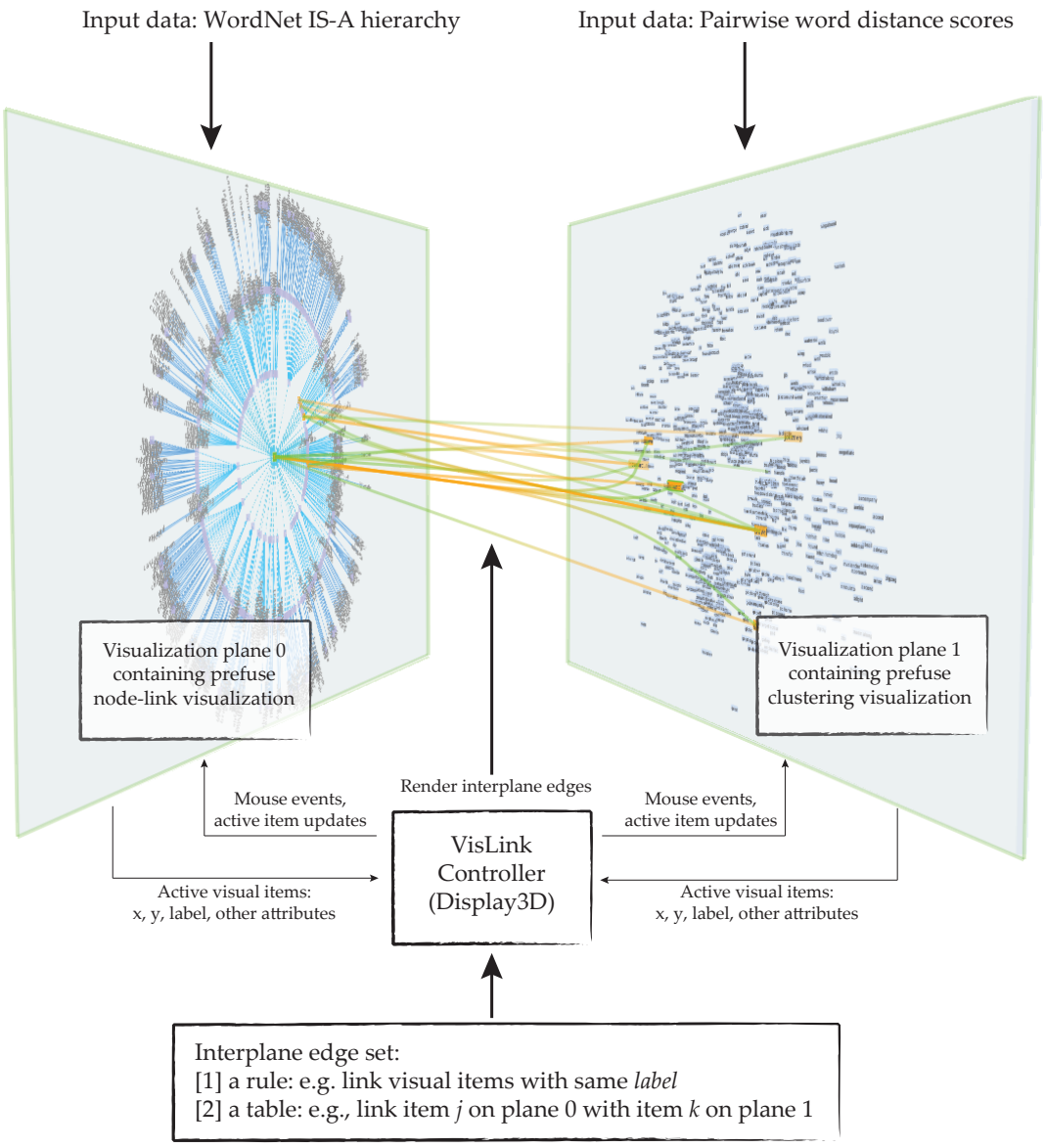


FIGURE 8.11: VisLink is implemented by connecting visualizations and a set of interplane edges through a controller, which directs input events and cross-plane queries, as well as rendering of visualization planes inter-plane edges.

among nodes with matching labels" (rules such as these must be translated into code that produces paired node indices). The connections between input data, the component visualizations, and the controller are illustrated in Figure 8.11.

The prefuse visualizations are shown on the visualization planes as textures, updated only when prefuse calls for a display repaint. Because the prefuse visualizations are drawn as textures on a 2D plane, VisLink could easily be extended to draw other shapes of visualization objects, such as cubes or spheres.

8.4 LINKING EXISTING VISUALIZATIONS

To demonstrate the ability of VisLink to add analytic power to existing prefuse-based visualizations, we used VisLink to bridge several of the demonstration applications that are distributed with the prefuse source code (Heer *et al.*, 2005) (with minor colour changes). Data on the occupations of members of the 109th Congress before election was mined from the Congressional Directory along with the zip codes they represent (Government Printing Office, 2008). This was combined with databases of zip code locations and fundraising totals of candidates in three recent federal elections, both provided with the prefuse distribution. We used three visualization planes and defined indirect relations between them.

First, a prefuse Treemap (Johnson and Shneiderman, 1991) was used to show the relative popularities of various occupations before election (see Figure 8.12, left). This was linked through the rule `Candidate had Occupation` to the prefuse-provided *¢ongress* visualization by Heer (2007). *¢ongress* is a scatterplot of individual fundraising success, ordered along the x-axis alphabetically by state of candidacy (Figure 8.12, center). This plot shows the candidates' party through node colour and whether they were running for the House or Senate through node shape. The y-axis shows fundraising success, and the range can be interactively altered with a slider (not shown in figure). This was linked to the prefuse reimplementation of the *zipdecode* (Fry, 2007) visualization of zip code geographic locations (Figure 8.12, right) through the rule `Candidate represents Zip Code`. Inter-plane edges link occupations to candidate nodes and candidates to map regions they now represent. Complex questions such as, "*Where did the most successful fundraising former journalist get elected?*" can be quickly answered. To implement this visualization, the bulk of the

FIGURE 8.12: VisLink was applied to bridge existing *prefuse* visualizations. Views of the constituent visualizations, from 2D equivalency mode, are shown along the bottom. The Treemap node 'journalist' is activated, propagating inter-plane edges to the scatterplot (showing journalists are not particularly outstanding fundraisers), and onward to the zip code regions that elected journalists now represent.

work came through creating and parsing the new database (occupations and zip codes) to generate inter-plane edges from our rules.

8.5 DISCUSSION

The VisLink technique offers a new way to look at the relationships amongst visualizations, but there remain several difficulties and unresolved issues for future research. The creation of a VisLink visualization starts with the selection of the constituent visualizations to compare. Making this selection — finding appropriate data and choosing appropriate representations — is as difficult within VisLink as it is in everyday visual analytics work, and may be best handled by data and visual experts. Some visualizations, such as node-link diagrams, seem to work better with inter-plane edges than others, such as Treemaps and other types of embedded hierarchy, where it is more difficult to see the connections to non-leaf nodes.

For visualizations with rich sets of inter-plane relations, the familiar spaghetti graph of edge congestion can quickly become a problem. Through bundling of edges, individual node activation, filtering techniques, and the ability to view the edge set from a series of angles, we have attempted to provide tools to handle this. However, additional techniques, for example edge lenses (Wong *et al.*, 2003) for 3D spaces, may improve the situation. The edge bundling technique we use works only for one-to-many edge sets. Many-to-many edge bundling as reported by Holten (2006) requires a hierarchical structure as an invisible backbone. In the datasets we used, such a structure was not available. However, since this work was completed, a promising solution which does not require a hierarchical backbone has been proposed (Holten and van Wijk, 2009) and may be a promising area for future development of VisLink.

Because VisLink contains any number of visualizations which may be pre-existing, the selection of colours for inter-plane edges is challenging. The orange-to-green colour scheme was selected because it interfered the least with the existing (predominantly blue) visualizations we imported into VisLink, and worked well both against a white background (for print) and a black background (on screen). However, orange-to-green is difficult to perceive for people with some forms of colour blindness. Inter-plane edge colouring will likely have to be customized to the constituent visualizations.

When working in a 3D space, issues of perspective must be considered. It is possible that perspective projection introduces a visual bias for closer regions of the planes and closer inter-plane edges. Directional bias may be introduced by the default views (side view presents bias toward vertical inter-plane patterns). 2D false symmetry effects may also occur. An analyst must be careful to view a VisLink visualization from several directions before drawing conclusions about apparent patterns in the data.

8.6 SUMMARY

In this chapter we have described VisLink, a visualization environment in which one can display multiple 2D visualizations, re-position and re-organize them in 3D, and display relationships between them by propagating edges from one visualization to another. Through reuse of the powerful spatial visual variable, we have introduced a method for visualizing multiple relations without any relation relinquishing its *spatial rights*.

The VisLink environment allows the viewer to query a given visualization in terms of a second visualization, using the structure in the second visualization to reveal new patterns within the first. By choosing a set of data items in visualization A and doing a one level propagation to visualization B, VisLink shows where items in A are related to items in B. Propagating the edges back again reflects the information gathered from visualization B to the structure of visualization A. Thus, using the example in Figure 8.9, starting from a similarity-based word visualization A, propagating edges from a chosen word into WordNet visualization B and back again reveals synonyms of the selected word in visualization A. Through spreading activation, bundled edges can be propagated between visualizations to any chosen depth.

VisLink displays multiple 2D visualizations on visualization planes while maintaining full 2D interactivity for each component visualization. 3D interaction widgets are provided to simplify 3D interaction and navigation. Relationships among visualizations can be revealed using methods such as selection and filtering for addressing edge congestion.

In the *space of linguistic visualization*, VisLink addresses the needs of linguistic, natural language processing (NLP), and CL research, is designed for expert analysts, and offers a interaction at the data-annotate level (add inter-plane edges) as well as adjustments in the visual mapping (choice

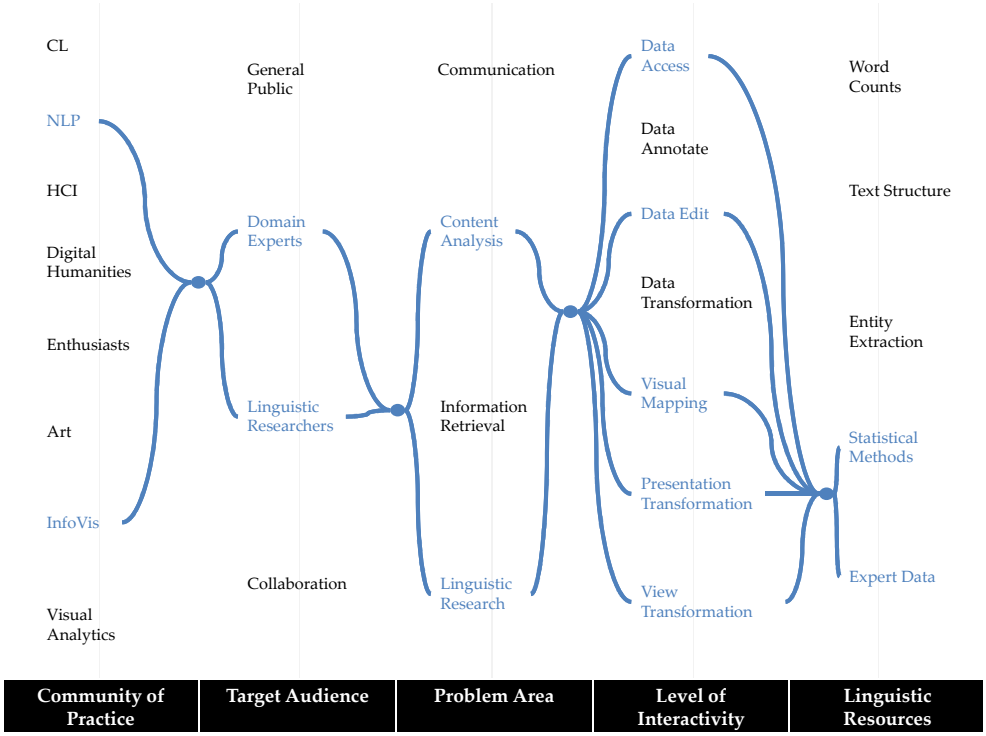


FIGURE 8.13: VisLink in the space of linguistic visualization.

of representation), presentation, and view. The characteristics of VisLink are summarized in Figure 8.13. VisLink is also a general technique, applicable outside the linguistic visualization. VisLink enables investigation of new types of linguistic problems through interactive visualization. For example, VisLink would enable instantiation of the model/signal mappings illustrated by Brighton and Kirby (2006) in order to understand the different types of linguistic evolution. We could also apply the system to additional lexical distance measures, to link clusterings over various measures. In this case, the inter-plane edge patterns would reveal whether 'nearness' by one distance measure is dependent on the other measure. Cross-visualization queries would allow for targeted explorations of the space.

VisLink can be a tool for bridging the through linking and providing interactive views over pre-existing information graphics. For example, a static visualization created by a CL could be imported into VisLink. The data items on the plane could be hand-annotated by clicking their position to create an invisible visual item at that location. Then the previously static graphic can be connected to other graphics or interactive visualizations with VisLink 3D edges.

Part V

CLOSING

9 CONCLUSIONS

Man is still the most extraordinary computer of all.

— John F. Kennedy

In the previous five chapters we have described five distinct design studies covering the problem areas of real-time communication, content analysis, and natural language processing research through directly linking highly interactive visualizations with natural language processing (NLP) algorithms and models of linguistic structure created by linguistic experts. In this chapter we will discuss some of the general linguistic visualization challenges that arose in more than one design study. We then review the contributions of this dissertation, leading into a discussion of possibilities for future research both leading from each design study and more generally for linguistic visualization. We conclude with some higher-level closing remarks.

9.1 CHALLENGES OF VISUALIZING LANGUAGE

In designing and implementing the five design studies discussed in this dissertation, some common challenges arose which, while shared by other visualizations which use text for labeling, were particularly noticeable due to the textual nature of the core data in linguistic visualization. Many of these challenges were previously discussed within the design studies, but we collect them here as a coherent set of common issues affecting more than one of our designs.

In the following discussion, we will identify the specific design studies involved as: Uncertainty Lattices UL (Chapter 4), DocuBurst DB (Chapter 5), Parallel Tag Clouds PTC (Chapter 6), Bubble Sets BS (Chapter 7), VisLink VL (Chapter 8).

9.1.1 *Legibility*

As the core data of our collection of design studies is textual, each of our five representations contain a significant amount of rendered text. The *legibility* of rendered text — the ability to *discern* the characters — is a complicated function, affected by anti-aliasing (Aten *et al.*, 2002), luminance contrast between text and background (Stone, 2003, p. 259), size and choice of font (Bringhurst, 1996), blur/semantic-depth-of-field (Kosara *et al.*, 2001), and text orientation (Grossman *et al.*, 2007; Larson *et al.*, 2000). In creating these visualizations, we identified possible legibility challenges related to these factors:

- small font size due to: screen real estate (UL, DB, VL); data-based scaling (DB, PTC)
- orientation in 2D (DB, BS, VL)
- orientation in 3D (VL)
- foreground/background luminance contrast due to text atop a colour-mapped background (UL, DB)
- overlapping text due to: screen real estate (BS, PTC); transparent overlay (VL)
- interference due to the proximity of blurry borders (UL)

In each of these cases, a design trade-off was necessary. For example, text size is sometimes small in DocuBurst and Parallel Tag Clouds because of wide-ranging data values, or outliers. Selecting a scaling function which offers enough separation amongst items can require that some items are illegible. Wattenberg and Viégas (2008) present evidence of viewer preference for illegible type rather than a symbolic representation of missing text. Related to legibility is *readability*, or the *ease* of reading text — readability challenges we identify include following paths through Uncertainty Lattices to read a sentence as a coherent unit. To address this, the Uncertainty Lattices layout facilitates left-to-right reading, and places the algorithm's most likely solution along the bottom.

9.1.2 *Text Scaling*

As discussed in Chapter 2, studies show human perception to be more accurate for estimating a quantity from a length when compared to an

FIGURE 9.1: A large score for a long word can result in words overlapping with neighbouring words, but the average case is more legible than if the text were scaled to disallow possible overlaps.

area. When using text size to encode quantitative data, a designer has a choice: scale by the area of the rendered word, or scale the text size (*e.g.*, in points). If an unbiased reading were possible, it would also be easier to read data from text size accurately (height of text = 1D length). However, scaling by text size can result in long words appearing disproportionately large. Conversely, scaling by total word area can result in selecting a very large text size for short words. Drawing on the findings of Bateman *et al.* (2008), who report text size to be the more accurate quantitative encoding for tag clouds, we scale by text size in both DocuBurst and Parallel Tag Clouds. This created a trade-off in Parallel Tag Clouds, however. In order to maximize the legibility of the average case, we had to set the base size to a level which would permit long words to overlap adjacent columns in rare cases (see Figure 9.1). Normalizing by the longest word and highest score resulted in almost all words being too small to read.

9.1.3 *Ambiguity of Language*

Several different types of ambiguity and uncertainty related to language arose in these design studies. Uncertainty Lattices was *designed* to portray the ambiguity within the translation outputs, but in other places ambiguity is not easily measured and displayed. Automatic word-sense disambiguation (WSD) is an active area of NLP research — some approaches use WordNet as both a resource and a target (*e. g.*, Banerjee and Pedersen, 2002). This means the glosses and relations in WordNet provide both the data used for disambiguation as well as the set of senses to choose from. The results to date are not promising: Banerjee and Pedersen (2002) achieve a 32% accuracy on the standard *SENSEVAL-2* WSD evaluations.

Achieving highly accurate automatic WSD at the granularity of WordNet senses would be difficult. Consider the word *cat* — if this occurs in a text, it most likely means a feline animal of some sort. However, WordNet includes *cat* in eight noun synsets and two verb synsets, listed in Appendix A. Note that the designers of WordNet place synset members in approximate rank order by frequency and only three of those eight synsets have 'cat' as the first member (some senses are quite obscure, even to a native English speaker).

In the absence of WSD, to ameliorate the effects of ambiguity in DocuBurst, we attempted several simple alternatives to distributing word counts evenly to all senses of a word: apply count only to the first word in synset, dividing by number of senses and distributing score evenly amongst synsets (to discount the impact of highly ambiguous terms), dividing the score across synsets in which the word appears using relative frequency information provided within WordNet (not available for very many words), using a linear drop-off function to reduce the weight assigned to lower-ranked synsets. None of the options are as useful as we expect full word sense disambiguation would be (but we use the last option). For example, examining a book about 'cats' with DocuBurst will reveal a small signal on the branch for {large vehicle} (*Caterpillar*, a brand of heavy equipment).

Word sense disambiguation is also a challenge faced by the viewers of interfaces such as Parallel Tag Clouds, which display words somewhat out-of-context ("somewhat" because at least the domain, legal cases, is known). Recall, our legal scholar expert saw 'ostrich' in a Parallel Tag Cloud and understood it in the correct way, where we assumed the bird

was being referenced (see §6.6). Beyond applying domain knowledge, the other ways to disambiguate problematic words in Parallel Tag Clouds include: providing interactive access to the underlying source text, so that the reader can perform the disambiguation in context, and providing keyword-in-context (KWIC) overviews in tooltips.

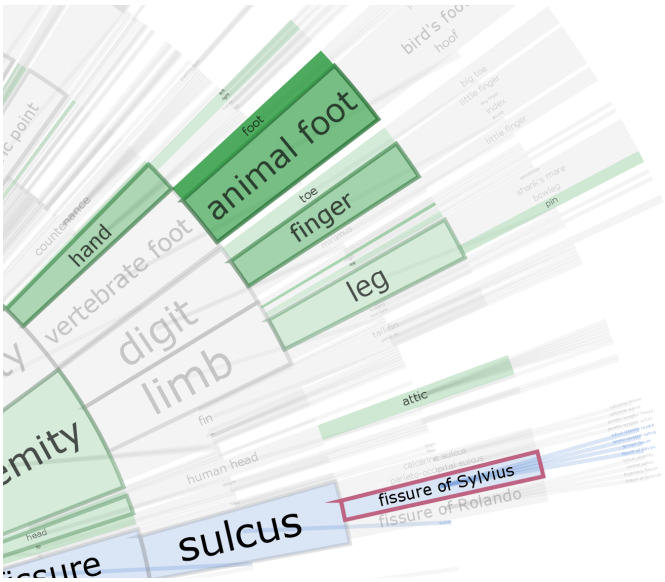
9.1.4 *Selecting an Appropriate Level of Abstraction*

When designing the content analysis visualizations DocuBurst and Parallel Tag Clouds, where complete overviews of the entire dataset (or even a significant fraction of it) was impractical without abstraction or filtering of the data (both for computational and perceptual reasons), it was challenging to automatically determine an appropriate level of abstraction.

In DocuBurst this manifested in two ways: first, the problem of knowing *where* to start an exploration of a particular text (*i.e.*, which root synset will make for an informative glyph?). Currently the analyst must specify a starting point. The second problem in DocuBurst is how to abstract the DocuBurst tree once it is created. For a great many synsets, the structure in the hyponomy tree is too large to visualize completely on standard computer hardware using DocuBurst. In the current implementation, we apply degree-of-interest (DOI) techniques to create distance-to-focus-node based filters over the tree (*i.e.*, collapse all nodes further than N steps away from the root or any selected focus node).

Abstracting on metrics such as distance-to-focus node falls into the category of *linguistically naïve*, as the filter does not take into account any of the semantics of the data. Ideally, the ontology itself would be designed in a manner to allow for normalized and comparable distance metrics (perhaps through a weighted hyponomy relation). Specifically, with WordNet, Hirschberg and Nakatani (1998) report examples of semantically inconsistent distances. From our own observation the level of specificity of a synset is only loosely related to the number of `IS-A` steps from the DocuBurst root. For example, "fissure of Sylvius: the deepest and most prominent of the cortical fissures" is at the same depth as foot, finger, and leg (see Figure 9.2a).

The specific selection of terms for the parallel tag clouds visualization is based on several filtering mechanisms, such as removing initial-uppers, the use of a frequency cut-off beam to ignore the most common (because they would not likely differentiate the facets) and most uncommon words



(a) The depth of a particular synset, and the number of steps in a chain of hyponomy links relating two concepts cannot be interpreted easily. Paths within WordNet composed of the same number of relations are often different semantic distance. Here we see {fissure of Sylvius} is at the same depth as {finger, leg, toe} in a DocuBurst view.

(b) The log likelihood technique for extracting significant terms selects terms of varying overall frequency and specificity, here specific terms 'ferritin' and 'misbehavior' appear in the same Parallel Tag Cloud with generic words 'here' and 'have'.

FIGURE 9.2: Level of abstraction challenges in linguistic visualization.

(because there are just too many of them), and a threshold on the log likelihood score. After all this processing, the top N words are selected to for each facet. The semantics of these words varies widely — some are quite specific while others are very generic (see Figure 9.2b).

9.1.5 *Handling Proper Nouns*

Within DocuBurst, proper nouns were simply ignored. As proper nouns generally do not appear in *WordNet*, they had no place within the structure of the DocuBurst tree. Proper nouns and the relationships between them carry important information. Integrating proper nouns into DocuBurst or in a linked, coordinated view would likely enhance the utility of the system.

In Uncertainty Lattices proper nouns were usually not translatable. In these cases, the translation model is abandoned in favour of a Web search for representative pictures. The only proper nouns to give recognizable results with this technique were place names and the names of famous individuals.

In Parallel Tag Clouds we excluded *initial uppers*, which caught most proper nouns due to the typographical conventions in English. We did this because the particular facet across which we divided the data (court division) could basically be classified by the proper nouns (place names and judges in the district). So, these names were all that would appear in the display.

9.2 SUMMARY OF CONTRIBUTIONS

This dissertation explored the space of closely linking NLP algorithms with information visualization (InfoVis). Our contributions fall into three main groupings: design studies, technical innovations, and general visualization concepts.

9.2.1 *Design Studies*

Through five design studies, we have introduced new techniques to couple interactive visualization with natural language processing and data:

LATTICE UNCERTAINTY VISUALIZATION exposes the *black box* of statistical machine translation (MT) and automatic speech recognition (ASR), to help people to make decisions about the quality of the outputs (Chapter 4).

DOCUBURST presents the first document content visualization spatialized using an expert-created linguistic ontology (Chapter 5).

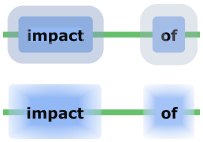
PARALLEL TAG CLOUDS contribute a method for detecting, displaying, and exploring differences within very large faceted text corpora (Chapter 6).

BUBBLE SETS address a specific and repetitive analysis task important to a group of MT researchers (Chapter 7).

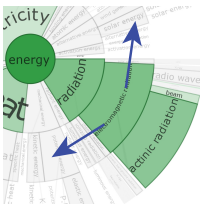
VISLINK provides a general platform within which multiple visualizations of language (or other data types) can be connected, cross-queried, and compared (Chapter 8).

9.2.2 *Technical Innovations*

In the development and implementation of the design studies, we have contributed eight technical innovations:



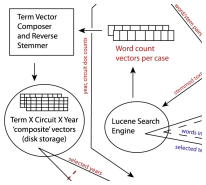
ENCODING UNCERTAINTY IN GRAPH NODES We explored the use of both graduated transparency and increasing degrees of boundary blurring as two possible methods for encoding uncertainty in the background of graph nodes. These techniques were designed to intentionally leave the center of the node clear for a text or other type of label (§4.5).



INTERACTING WITH RADIAL SPACE-FILLING TREES We contribute the interaction technique of adjusting the angular width of subtrees in RSF layouts using the mouse wheel while retaining the availability of the other buttons for simultaneous click+drag operations such as pan and zoom (§5.5.3). The source code for our radial space-filling (RSF) Tree layout and interaction is available from `http://www.christophercollins.ca/research/docuburst` and has been used in published extensions to the technique (*e.g.*, Byrne *et al.*, 2007; Hancock *et al.*, 2009; Pan and Wang, 2009).

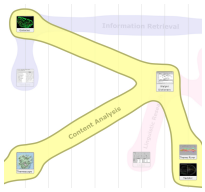
INTERACTIVE STUB EDGES TO INDICATE DISTANT CONNECTIONS In order to hint at the presence of distant connections between items in a visualization, we developed edge stubs: edges which are opaque near their connections but fade to transparency at a distance. The edges use a three-state interaction model: edges change to full opacity when the mouse pointer brushes over either endpoint and remain opaque if an endpoint is clicked. Groups of edges can also be activated to reveal patterns of connections.

COMPOSING AND CACHING TERM VECTORS In order to quickly create lists of significant terms based on any arbitrary subset of a corpus, we developed a method for precomputing, storing, and dynamically composing lists of significant words extracted from large faceted text corpora at interactive speeds (§6.5).

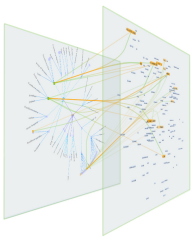


EDGE-ROUTING ALGORITHM TO MAINTAIN SET CONNECTEDNESS To create a skeleton for the implicit surfaces used in Chapter 7, we developed an edge-routing algorithm which subdivides edges to move them around any obstacles which are blocking the shortest path between between set members. The algorithm recalculates edge routes on realistic datasets at interactive speeds (§7.3).

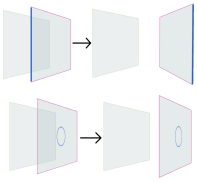
SET MEMBERSHIP ASSIGNMENT AT A DISTANCE Interfaces which use implicit surfaces to group objects commonly base membership on proximity. We contribute a method for adding items to a set through first activating a set then selecting the objects to add or remove. The set boundary flows around obstacles to connect members in arbitrary layouts.



RELATING MULTIPLE 2D VISUALIZATIONS IN 3D SPACE We contribute a general technique for re-using the spatial visual variable, making it possible to reveal relationships, patterns, and connections between two or more primary 2D visualizations within a constrained navigation 3D space (§8.2).



INTERACTION WIDGETS FOR MANIPULATING 2D PLANES IN 3D SPACE In order to facilitate changing the relative alignment of 2D planes in 3D environment using a 2D mouse, we create the *garage door*, *accordian*, and



book widgets which provide constrained interaction leading to preferred views (§8.2.2).

9.2.3 *Visualization Concepts*

Arising from this dissertation are three high-level conceptual contributions:

Spatial Rights

The use of *spatial rights* in this work is a new way of framing the primacy of the spatial visual variable. Used in Chapters 4, 7, and 8, this concept can be used as a reminder for visualization designers to consider which data dimension or dimensions should be given spatial rights. Making this decision then provides helpful constraints to guide further design.

Formalism for Multiple Relationship Visualizations

Visualizations of datasets containing multiple types of relations are increasingly common. In Section 8.1 we provide a formalism through which one can describe how spatial rights are preserved or infringed upon when more than one data relation is visualized. This conceptual framework allows for differentiation amongst existing approaches which employ multiple relations, multiple visualization types, and potentially coordinated views.

Cross-Visualization Query Technique

One of the primary purposes of creating a visualization for a data set is to query the data for specific details. Bertini *et al.* (2007) has categorized the types of queries that a visualization commonly supports. These include a range of queries from details about a given data item to comparative questions between pairs of data items to trends over subsets of data items to overviews of the dataset. All of these types of queries are within a single visualization. Visualizations in themselves can encode a great variety of types of relations within the data. Through VisLink, we provide the opportunity to visually formulate queries that make use of more than one visualization simultaneously. These visual queries are represented by connections between visualizations using bundled edges to link related

data items in the visualizations that match the query. Patterns in the cross-visualization edges may allow an analyst to see secondary relations between representations. Use of cross-visualization queries opens the door to expanding the expressive power of representations by connecting already existing visualizations (§8.2.3, §8.2.4).

9.3 BEYOND WORD COUNTS: FUTURE RESEARCH OPPORTUNITIES

Each of our design studies has inspired ideas for future research directions. You may notice that some of these have generated extensive ideas, resulting from interest within the research community. Many of the ideas for future directions discussed below arise from the challenges we identify in Section 9.1.

9.3.1 *Communication*

Our Uncertainty Lattices visualization relies on embedding of uncertainties on the nodes. Some statistical processing algorithms also provide scores for edges. Extending the visualization to incorporate edge uncertainties is a natural next step. Additionally, the hybrid layout algorithm used in this work does not reorganize the layout based on the viewer's selection. A more generic graph layout algorithm may allow for smoothly animating the currently selected best path to the bottom row of the lattice. This would enhance readability and reduce the jitter produced by the use of a force-directed algorithm. One potentially useful layout algorithm is the enhanced *Sugiyama layout*.

While our visualization of real-time communication is tightly coupled to two types of statistical NLP algorithms, the information flow follows a transactional model with no feedback component: the interface receives input, sends it to the NLP algorithm, which produces a set of results embedded in a lattice which is returned to the visualization for display. After this, the viewer may correct the translation or transcription using the interactive interface. The correction goes to the log, but is not fed back to the statistical model. The viewer-generated corrections could be informative for refining the statistical model.

Informal demonstrations of the Uncertainty Lattices system generated enthusiastic feedback. If cross-language visualized instant messaging was

provided as a service on the Web, the collection of anonymized records of corrective actions could quickly grow to become a significant resource.

Beyond the specifics of the Uncertainty Lattices visualization, opportunities to visualize communication include exploring personal information management: can closer coupling of NLP and InfoVis ease feelings of *information overload* felt due to overflowing e-mail inboxes and unchecked RSS feeds?

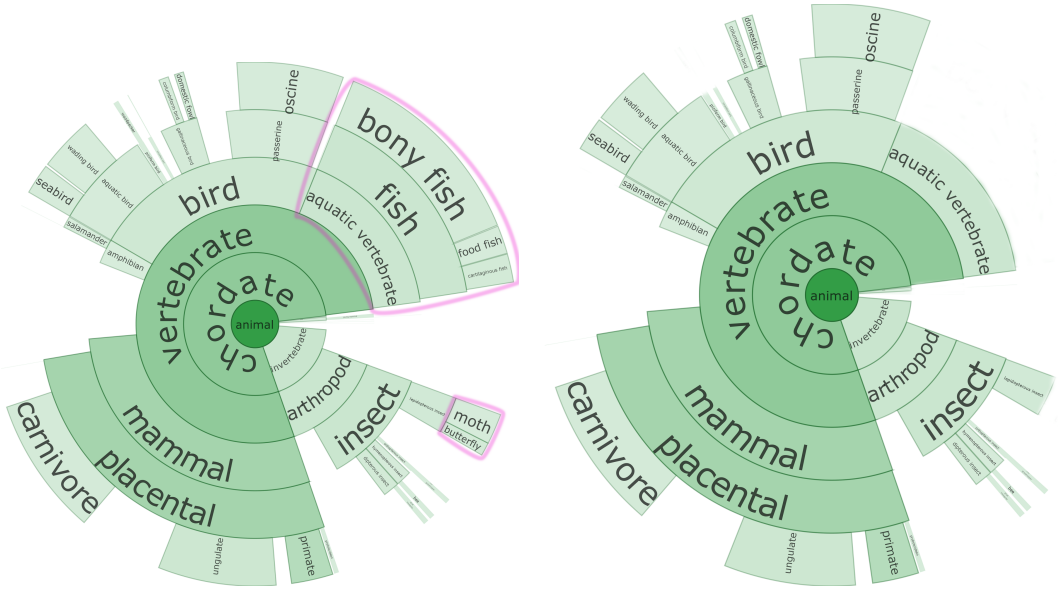
9.3.2 *Content Analysis*

DocuBurst

Initially motivated by the current lack of a digital equivalent of flipping through a book, this work leads well into an investigation of the DocuBurst technique to view the differences between two or more documents, which may be useful for plagiarism detection, document categorization, and authorship attribution. Existing digital library interfaces could be enhanced with arrays of DocuBurst glyphs, allowing comparison against one another or a baseline reference corpus to portray content in more pleasing and information-rich ways.

As previously discussed in Section 9.1, there are several common challenges which arise when designing linguistic visualizations. Here we divide ideas for future work to match the previously explained challenge it would address.

SCALABILITY From a data perspective, the original goal of creating a structured view of which parts of *an entire language* are included in a document, merits further research. Recall that DocuBurst only encompasses nouns and verbs within WordNet. As with all text visualizations, it is necessary to view a subset of language due to limited display space and computational resources with extremely large data. Views rooted at {entity} and covering all English nouns appear cluttered and interaction is too slow for comfortable use. It is commonly held that WordNet sense-divisions are too fine-grained for many computational applications; investigation into other ways to abstract WordNet or application of the technique to an alternative ontology may help alleviate this problem. It may also be advantageous to use more advanced rendering techniques to render clusters of tiny, unreadable nodes as a single shape.



(a) Even (distance-based) tree cut, with zero-count nodes removed. Nodes highlighted in pink would be candidated for removal by an uneven tree cut.

(b) Sketch of uneven tree cut based on distance from central focus and the underlying data profile. For example, as the vast majority of the {aquatic vertebrate} synset is present under {fish}, we abstract to the most generic term without losing much information. But, as {salamander} is present under {amphibian} while the {frog} subtree is not, we leave the more specific synset.

FIGURE 9.3: Comparing even and ideas for uneven tree cut methods to abstract DocuBurst.

IMPROVED AND UNEVEN ABSTRACTION The goal here is to create a method which will abstract away unnecessary detail to clarify the view while retaining salient information. Beyond simple distance measures, it would be useful to be able to abstract an uneven ontology like WordNet with an *uneven tree cut*. Such a tree cut could be based on the word counts gathered from the document as well as the structure of the hyponymy tree.

First, we need appropriate ways to abstract and filter the data. We propose to explore uneven tree-cut models as the scoring function used by the DOI filter. As a starting point, the score value would be a function of both the structure of the tree and the meta data (*e. g.*, word counts) assigned to nodes of the tree. For example, if a node has five children, and four of them have significant non-zero occurrence counts, we may propagate the counts from the children to the parent and remove the children. This is because of the transitive nature of the `IS-A` relation in WordNet. As a concrete example: if the synset {fish} has three member synsets, {bony fish}, {food fish}, {cartilaginous fish}, and all have significant occurrence counts, we may safely remove the members, propagating their occurrence counts up to {fish}, because generally the document is about {fish}. Alternatively, if the synset {amphibian} has two member synsets, {salamander} and {frog}, and *only* {salamander} has a non-zero occurrence count, we cannot assume the document to be about {amphibians} generally and must retain the node {salamander}. This is illustrated in Figure 9.3.

SUGGESTING STARTING POINTS Finding a place to begin exploration is another challenge with the current implementation. Providing hints for which synsets may be of interest as visualization roots for a particular document or set of documents may assist an analyst to find views of interest. Methods which may be useful include suggesting synsets with a high fraction of non-zero leaves below them, synsets with an unusual pattern of non-zero nodes below them, or synsets for which the cumulative count divided by the number of leaves is high, indicating an area of unusual concentration.

ADDITIONAL SCORING FUNCTIONS Currently word occurrence count is the only available word scoring function. Other scoring functions, such as the log-likelihood ratio (Dunning, 1993), could be used to highlight important or unusual words in a document. Other text features, such as

hapax legomena (words which occur only once) could be used with the WordNet-based layout to provide special-purpose summaries of content.

VISUAL ENCODINGS Visually, the use of transparency to indicate word occurrence creates an intuitive mapping between data and visual appearance. However, it also introduces the possibility of misleading illusions. For instance, siblings in DocuBurst are unordered; furthermore, non-sibling nodes may be adjacent. By chance, unrelated nodes that both have high occurrence counts can appear as a large swath of strong colour. Gestalt perception may lead viewers to impart significance to this coincidence. Stronger node borders would distinguish these regions, but node borders become obstructive on small nodes. Finding an experimentally validated solution to this design trade-off could impact space-filling visualizations in general.

Parallel Tag Clouds

CLARIFYING WORD SENSES It may be possible to partially disambiguate words by re-organizing the vertical arrangement of the layout to cluster words into co-occurrence-based sets. If 'bank' and 'river' appear the same set, one would assume the financial institution sense of 'bank' is unlikely. Such clustering could alternatively be visually achieved by adding an additional dimension of visual encoding such as encoding set relations with hue. Clusters could also be displayed overParallel Tag Clouds without reorganizing the alphabetic layout by using Bubble Sets.

AUTOMATIC REORGANIZATION OF COLUMNS The ordering of axes is an important factor when designing a parallel coordinates view. In this work, we took the approach that the data contains a semantic relation (the ordering of the circuits from First to Eleventh). Disrupting semantically meaningful arrangements is potentially problematic (Misue *et al.*, 1995). For other data sets, automatic column reordering may be appropriate, or a facility for interactive reordering could be provided. If the facets have no natural order, then the columns of the visualization could be rearranged interactively or automatically (*e.g.*, Inselberg and Dimsdale, 1990) to enhance the probability that related columns would be adjacent.

REVEALING CHANGE Our instantiation of Parallel Tag Clouds includes change highlighting through colour. Additional methods to reveal change

are needed, particularly to reveal which terms are removed from view when a parameter changes.

IMPROVING SPEEDS AT RUN-TIME Accomplishing view changes at interactive speeds presents a computational challenge. While in our initial prototype we pre-processed the court data into year-long chunks, there remained some lag in loading a selected time period. During this lag, word count by facet arrays are loaded for each year, then composed to a single vector. The assignment of G^2 scores requires a single frequency vector computed over the selected subset of the data. This prevents pre-calculation of the G^2 score as it would need to be calculated over the power set of data subsets (in our case, time ranges). This also prevents extensive filtering of the word set in the pre-processing stage, as it is difficult to predict which words may have a significant G^2 score when examining only frequency data. We apply some well-motivated heuristic approaches to reducing the lexicon at the pre-processing stage, but these methods are approximate and may introduce errors.

On-line calculation of the significance score results results in a bottleneck in the pipeline at runtime: load operations (slow) are followed by scoring (slow), filtering, representational transformations, presentation transformations, and the creation of a view. New ways to pre-process large corpora for fast loading of arbitrary subsets of data, or new scoring methods which can be calculated quickly or (better) in advance will improve the capacity of this method to handle larger corpora.

9.3.3 *Linguistic, NLP, and CL Research*

Bubble Sets

Within our isocontour approach we have implemented several heuristics to reduce surface calculation and rendering time, such as grouping pixels for potential calculations and restricting the regions in which items influence the potential field. The current implementation works without noticeable lag (items can be dragged and the surface follows) for our examples (order of 100 nodes, 10–20 sets). However, as the number of items, the screen resolution, or the number of sets increases, so will the rendering time. Additional techniques, such as grouping close items into larger

pseudo-nodes, and caching the energy field values between frames may increase the capacity of the system.

VisLink

APPLICATIONS We have described VisLink primarily with examples from a single data set. VisLink may be applicable to a rich set of problems in linguistic data analysis. For example, using different lexical distance measures to organize words on VisLink planes, it would be interesting to observe the patterns of inter-plane edge connections as we did with the simple spelling-related similarity measure. Other possible applications include relating parse tree representations, comparing different ontologies and structured knowledge sources, and investigating language change over time. It may also be useful to relate linguistic and non-linguistic data with VisLink, and we have been approached by researchers from large firms such as DOW Chemical to investigate relating data such as chemical diagrams with linguistic information such as patent texts. Opportunities also exist to expand the capabilities of inter-representational queries, for example, by providing for a rich query language that can filter each visualization plane separately.

LINKING INFORMATION GRAPHICS VisLink can be a tool for bridging the through linking and providing interactive views over pre-existing information graphics. For example, a static visualization created by a computational linguistics (CL) could be imported into VisLink. The data items on the plane could be hand-annotated by clicking their position to create an invisible visual item at that location. Then the previously static graphic can be connected to other graphics or interactive visualizations with VisLink 3D edges. In future work we plan to create an easy-to-use import utility to import, scale and position, and annotate information graphics with visual items and inter-plane edges.

MANAGING EDGE CONGESTION VisLink suffers from sometimes revealing an unwieldy set of 3D edges when an entire plane is activated. Depending on connection patterns, the inter-plane edges can even grow quite large through query propagation. Future research will investigate techniques for managing edge congestion, such as 3D edge bundling techniques, the use of interaction tools to isolate edge sets of interest, and stepwise animation of the propagation of cross-plane activation.

Exploratory Data Analysis

An area of *linguistic visualization* within the problem area of linguistic, CL, and NLP research for which visualization offers much promise is *exploratory data analysis*. One type of exploratory analysis is visualizing corpora, either through overviews or discrete document views. This is a specialized type of content analysis targeted at understanding linguistic data. Interactive exploratory techniques could be used for quality control of a corpus, to deeply investigate patterns of inter-annotator disagreement, and to discover areas of imbalanced coverage. Exploratory data analysis could also take the form of algorithm visualizations, providing insight into 'black box' models through visualizing variance in model predictions, *e.g.*, "What changes when I adjust this parameter?".

Understanding NLP Processes

Our work on Uncertainty Lattices and Bubble Sets began an investigation into visualization to aid in understanding NLP processes. This direction has a lot of avenues to explore, including algorithmic visualization of automata (as they are running), visualizing non-determinism, visualizing dynamic programming processes such as chart pruning and beam search, and tracking the search space of hypotheses in statistical models such as MT and ASR.

9.4 CLOSING REMARKS

Despite the improvements in NLP algorithms and the increase in computing power, humans are still clearly needed in the linguistic analysis process. The joys of human language — the ability to recombine words to produce new meaning, to say one thing yet mean another, to poetically use metaphors without even noticing, to say a very few words and have friends know your intent — these are the very reasons computers cannot replace us. But given the *information society* we live in, we need new tools and techniques to manage and comprehend all the linguistic data we are producing, sharing, and archiving. *Information overload*, *information anxiety*, and *information addiction* (Bawden and Robinson, 2009) are becoming more common phrases in our vocabulary. Just as representing numbers with pen and paper acts as a cognitive aid for processing long division, perhaps new forms of cognitive aids are possible to increase

our capacity to consume linguistic data. Computers are exceedingly fast when faced with bulk data processing; humans are nuanced at interpreting meaning. This dissertation explores the possibility that interactive linguistic visualization, grounded in state-of-the art natural language processing, may produce new forms of human-computer optimized systems to collaboratively analyze linguistic data.

To understand the how the growing number of linguistic visualizations relate to one another, and to the design studies presented in this dissertation, we frame of the *space of linguistic visualization* along five dimensions: the community of practice creating the visualization, the target audience, the problem area, the level of interactivity provided by the interface, and the type of linguistic resources used. We also introduce the concept of a *linguistic visualization divide* — The gulf separating sophisticated natural language processing algorithms and data structures from state-of-the-art interactive visualization design..

A greater number of close collaborations between researchers and designers with expertise in CL, NLP, and InfoVis would help us move towards closing this divide. Two current factors are facilitating an increasingly fast movement in this direction. First, there is a greater focus generally on improving support for interdisciplinary collaborations. Second, many other disciplines, such as visual analytics and digital humanities, are beginning to draw on techniques from both NLP and InfoVis in their own research processes. So, it may be that the innovations that close the divide come from communities of practice outside the two research fields that primarily inform this work.

Additionally, the Web is opening up the space of innovation to provide for rapid prototyping and wide dissemination of visualizations using open application programming interfaces (APIs) for data provision and commonly available protocols for visualization display. What is missing in the realm of software-as-a-service computing are the open APIs for NLP. The general lack of sophistication in the NLP which is linked to linguistic visualizations may be because NLP algorithms are perceived as unreliable, difficult to interpret, resource intensive, and complex to implement. The provision of open APIs for NLP would address these issues. First, if such APIs are developed, they should not present their results as black box solutions, but allow for access to metadata about the NLP, such as confidence scores. This reliability information can be used in information visualizations to assist people in interpreting the outputs of NLP. Second,

on interpretation, the API could provide approachable and clear documentation about how the outputs are calculated, then, through good design, visualizations could make interpretation even easier. Third, by providing software-as-a-service over the Web, resource-intensive algorithms such as translation, sentiment analysis, summarization, and keyword detection could be parallelized and calculated 'in the cloud'. Restricted computing resources for NLP within the browser setting was the most challenging issue blocking deployment of the Uncertainty Lattices, DocuBurst, and Parallel Tag Cloud design studies as Web applications. Fourth, by providing NLP algorithms as a service, the complexity of implementation issues are removed. A developer could rely on the underlying NLP to be state-of-the-art, while knowing the interface to the visualization will not change.

The media we use for communication, representation, and storage of linguistic information are always evolving. In an information society faced with growing problems of information overload, development of new external cognitive aids in the form of linguistic visualizations may be a practical next step in this evolution.

APPENDICES

A

WORDNET SENSES OF CAT

1. (n) Synonyms: cat, true cat
   Sense: feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats
2. (n) Synonyms: guy, cat, hombre, bozo
   Sense: an informal term for a youth or man; "a nice guy"; "the guy's only doing it for some doll"
3. (n) Synonyms: cat
   Sense: a spiteful woman gossip; "what a cat she is!"
4. (n) Synonyms: kat, khat, qat, quat, cat, Arabian tea, African tea
   Sense: the leaves of the shrub Catha edulis which are chewed like tobacco or used to make tea; has the effect of a euphoric stimulant; "in Yemen kat is used daily by 85
5. (n) Synonyms: cat-o'-nine-tails, cat
   Sense: a whip with nine knotted cords; "Sailors feared the cat"
6. (n) Synonyms: Caterpillar, cat
   Sense: a large tracked vehicle that is propelled by two endless metal belts; frequently used for moving earth in construction and farm work
7. (n) Synonyms: big cat, cat
   Sense: any of several large cats typically able to roar and living in the wild
8. (n) Synonyms: computerized tomography, computed tomography, CT, computed axial tomography, CAT
   Sense: a method of examining body organs by scanning them with X rays and using a computer to construct a series of cross-sectional scans along a single axis
9. (v) Synonyms: cat
   Sense: beat with a cat-o'-nine-tails
10. (v) Synonyms: vomit, vomit up, purge, cast, sick, cat, ...
    Sense: eject the contents of the stomach through the mouth; ...

B

BUBBLE TREE DTD

```
<!--
DTD describing a bubble tree containing a tree & a list of bubbles over nodes.

Inspired by the TreeML DTD by Jean-Daniel Fekete and Catherine Plaisant
See http://www.cs.umd.edu/hcil/iv03contest/index.shtml
-->

<!ELEMENT bubbleTree (attribute*,bubbleType*,bubbleDeclarations?, tree, fsentence?, bubbleEdgeList?)>
<!ATTLIST bubbleTree
   label CDATA #IMPLIED
>

<!-- e.g. <bubbleType id="reorder" color="102">re-order rule</bubbleType> -->
<!ELEMENT bubbleType (#PCDATA)>
<!ATTLIST bubbleType
    id ID #REQUIRED
    colour CDATA #IMPLIED>

<!ELEMENT bubbleDeclarations (declarations?, bubbleDecl+) >

<!-- declarations will specify the names & types of attributes available for each node -->
<!ELEMENT declarations (attributeDecl+) >
<!ELEMENT attributeDecl EMPTY>
<!ATTLIST attributeDecl
  name CDATA #REQUIRED
  type (Integer|Long|Float|Double|String|Date|Category) "Integer"
>

<!-- comments about a bubble or any part of the tree -->
<!ELEMENT comments (comment+)>
<!ELEMENT comment (#PCDATA)>
<!ATTLIST comment
    author CDATA #IMPLIED
    date CDATA #IMPLIED
    score CDATA #IMPLIED
>

<!ELEMENT bubbleDecl (attribute*,comments?)>
<!-- id must be unique for each bubble, set bubbletypeid to
        distinguish tree bubbles from sentence bubbles -->
<!ATTLIST bubbleDecl
```

```
  id ID #REQUIRED
  bubbletypeid IDREF #IMPLIED>
<!-- e.g. <bubbleDeclarations><bubbleDecl id="1" bubbletypeid="reorder">
                <attribute name="rule number" value="2203" /></bubbleDecl></bubbleDeclarations> -->

<!-- TREE -->

<!ELEMENT tree (declarations?, (branch | leaf)* )>

<!-- nodes can appear in multiple bubbles;
        not present in current data, but allowed for generalizability -->
<!ELEMENT branch (attribute*, bubbles+, comments?, (branch | leaf)+ ) >
<!ATTLIST branch label CDATA #REQUIRED>

<!ELEMENT bubbles EMPTY>
<!ATTLIST bubbles
  memberof IDREFS #REQUIRED>

<!ELEMENT leaf (attribute*, comments?, bubbles+)>
<!ATTLIST leaf label CDATA #REQUIRED>

<!-- names and types of values should match declarations -->
<!ELEMENT attribute EMPTY >
<!ATTLIST attribute
  name CDATA #REQUIRED
  value CDATA #REQUIRED >
```

C

PARALLEL TAG CLOUDS INTER-COLUMN EDGE DESIGNS

In the process of developing the final design for Parallel Tag Cloud inter-column edges, we investigated several alternatives. Ideally, the edges should:

- reveal patterns of connections between columns;
- show where a particular word appears more than once in a plot;
- not interfere with reading the words;
- be traceable over long distances;
- not overload the visualization (perceptually or aesthetically).

The following figures explore the pros and cons of some of the alternative designs considered.

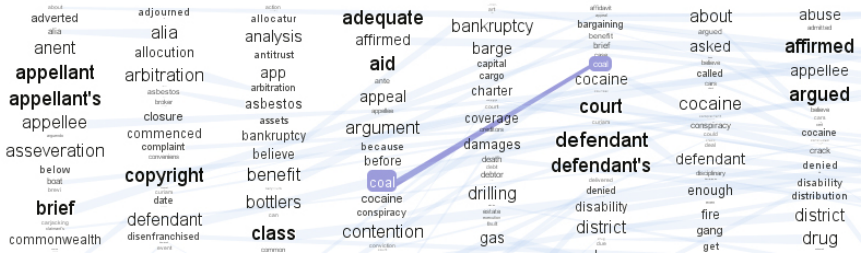


FIGURE C.1: Semi-transparent edges connect words across columns and become opaque when selected. Advantage: reveals patterns in edges. Problems: interferes with legibility of words, large volume of edges makes tracing edges difficult.

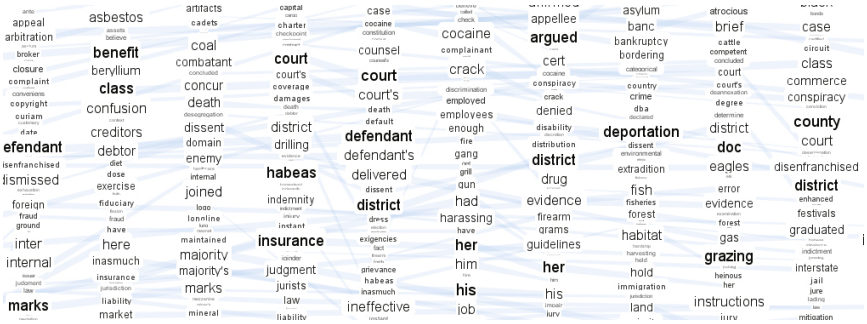


FIGURE C.2: An opaque background on words places the semi-transparent edges behind. Advantage: improves legibility over C.1. Problem: more difficult to follow edges.

FIGURE C.3: Word hue indicates the presence of an edge. Edges are only shown on hover or select. Advantage: edges do not interfere with legibility, grouping of words with edges are visible in columns. Problems: hue variance may reduce legibility of words, different hues may have different perceived weight, no indication of edge direction or distance.

FIGURE C.4: A dot beside a wor indicates the presence of an edge. Edges are shown only on hover or select. Advantage: edges to not interfere with legibility. Problems: dots are difficult to see, dots do not indicate direction of edge, patterns of connections not visible.

FIGURE C.5: The colour of an edge stub indicates the column it is connected to. Words are coloured according to their column. Complete edges are only shown on hover or select. Advantages: edges do not interfere with legibility, hue and stub direction indicate destintion. Problems: use of many hues creates an overwhelming visual effect, cognitive load of tracing hue from edge to column is high, text hue may compromise legibility.

FIGURE C.6: The colour of an edge stub indicates the column it is connected to. Column backgrounds are filled in column-specific hue. Complete edges are only shown on hover or select. Advantages: edges do not interfere with legibility, hue and stub direction indicate destintion, column background hue clearer than using text hue. Problems: use of many hues creates an overwhelming visual effect that is difficult to look at, cognitive load of tracing hue from edge to column is high, coloured background does not extend to fit text, which may compromise legibility.

FIGURE C.7: The colour of an edge stub indicates the column it is connected to. Word backgrounds are filled in column-specific hue. Complete edges are only shown on hover or select. Advantages: edges do not interfere with legibility, hue and stub direction indicate destintion, word background is clearer than using text hue. Problems: use of many hues creates an overwhelming visual effect, cognitive load of tracing hue from edge to column is high.

FIGURE C.8: The random colour of a wedge matches the colour of the other end of the edge. Wedges would meet at a point at destination if extended, using the technique of (Gustafson *et al.*, 2008). Complete edges are only shown on hover or select. Advantages: wedges to not interfere with legibility, hue can be matched to find edge connections, destination of wedge can be extrapolated by extending edges to meet at a point. Problems: discerning hue on a wedge border is difficult due to small area, extrapolating destination point for wedge increases cognitive load.

FIGURE C.9: The random colour of a stub matches the colour of the other end of the edge. Stub width at each end corresponds to word height. Complete edges are only shown on hover or select. Advantages: wedges to not interfere with legibility, hue can be matched to find edge connections. Problems: discerning hue on a stub border is difficult due to small area.

FIGURE C.10: The colour of an edge stub indicates the column it is connected to. The column labels are filled in a column-specific hue. Complete edges are only shown on hover or select. Advantages: edges to not interfere with legibility, hue and stub direction indicate destination, colouring of column label avoids negative effects on legibility. Problems: use of many hues creates an overwhelming visual effect, referencing stub colour and column label colour requires repetitive visual search.

FIGURE C.11: Edge stubs are drawn in a single hue. Complete edges are only shown on hover or select. Advantages: edges to not interfere with legibility, stub direction indicates destination, stub shape indicates relative size of words on each end of edge, simple colouring creates a pleasing aesthetic. Problems: without hover or select, determining the destination column for a stub is not possible. This is the version used in the final prototype presented in Chapter 6.

BIBLIOGRAPHY

Joshua Albrecht, Rebecca Hwa, and G. Elisabeta Marai. 2009. The Chinese room: Visualization and interaction to understand and correct ambiguous machine translation. *Computer Graphics Forum (Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis))*, 28(3). *Cited on page 109.*

Keith Alcock. 2004. WordNet relationship browser [online, cited 20 February, 2006]. Available from: `http://www.ultrasw.com/alcock/`. *Cited on page 127.*

Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. 2009. Revisiting read wear: Analysis, design, and evaluation of a footprints scrollbar. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI). Cited on page 49.*

Alexander Villacorta Alexander, Karl Grossner, Jonathan Ventura, Anne-Marie Hansen, Emily Moxley, Joriz De Guzman, and Matt Peterson. 2007. Spheres of influence. In *SIGGRAPH 2007 Art Gallery*, page 254. ACM. *Cited on page 66.*

Robert Amar and John Stasko. 2004. A knowledge task-based framework for design and evaluation of information visualizations. In *Proc. of the IEEE Symp. on Information Visualization*, pages 143–149. IEEE, October. *Cited on pages 14, 38, and 100.*

Keith Andrews and Helmut Heidegger. 1998. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proc. of IEEE Symp. on Information Visualization (InfoVis), Late Breaking Hot Topics*, pages 9–12. IEEE. *Cited on page 126.*

Caroline Appert and Jean-Daniel Fekete. 2006. Orthozoom scroller: 1D multi-scale navigation. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*. ACM, April. *Cited on page 49.*

Thomas R. Aten, Leo Gugerty, and Richard A. Tyrrell. 2002. Legibility of words rendered using ClearType. In *Proc. of the Annual Meeting of the Human Factors and Ergonomics Society*, pages 1684–1687. HFES. *Cited on page 238.*

Michael Balzer and Oliver Deussen. 2007. Level-of-detail visualization of clustered graph layouts. In *Proc. of the Asia-Pacific Symposium on Visualization (APVis)*, pages 133–140. IEEE Computer Society. *Cited on page 183.*

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proc. of the Third Int. Conf. on Intelligent Text Processing and Computational Linguistics*, pages 136–145. *Cited on page 240.*

C. Barrière and C. St-Jacques. 2005. Semantic context visualization to promote vocabulary learning. In *Proc. of the Joint Annual Meeting of the Assoc. for Computers in the Humanities & Assoc. for Literary and Linguistic Computing*, pages 10–12. *Cited on page 214.*

Scott Bateman, Carl Gutwin, and Miguel Nacenta. 2008. Seeing things in the clouds: The effect of visual features on tag cloud selections. In *Proc. of the ACM Conf. on Hypertext and Hypermedia*. ACM. *Cited on pages 148 and 239.*

G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. 1999. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall. *Cited on page 214.*

Daniel Bauer, Pierre Fastrez, and Jim Hollan. 2005. Spatial tools for managing personal information collections. In *Proc. of the Hawaii Int. Conf. on System Sciences (HICSS)*, page 104.2. *Cited on pages 200 and 203.*

David Bawden and Lyn Robinson. 2009. The dark side of information: overload, anxiety, and other paradoxes and pathologies. *Journal of Information Science*, 35. *Cited on page 254.*

Ben Bederson. 2000. Fisheye menus. In *Proc. of the ACM Symp. on User Interface Software and Technology (UIST)*, pages 217–226. ACM. *Cited on page 134.*

Jacques Bertin. 1983. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press. *Cited on pages 14, 26, 27, 31, 32, 33, 81, 101, 104, 107, and 129.*

Enrico Bertini, Catherine Plaisant, and Giuseppe Santucci. 2007. Beyond time and errors; novel evaluation methods for information visualization. *Interactions*, pages 59–60, May–June. *Cited on page 246.*

Enrico Bertini and Giuseppe Santucci. 2006. Visual quality metrics. In *Proc. of the Conference on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. *Cited on page 39.*

Ryan Bigge. 2007. You are looking at an open book: Docuburst. *The Toronto Star*, June 10. *Cited on page 138.*

J. F. Blinn. 1982. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256. *Cited on page 186.*

D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. 2005. *3D User Interfaces*. Addison-Wesley. *Cited on page 217.*

John Bowring, editor. 1843. *The Works of Jeremy Bentham*, volume 7, page 282. Thoemmes Continuum. *Cited on page 141.*

Cindy Brewer and Mark Harrower. 2002. Colorbrewer. Available from: `http://www.colorbrewer.org`. *Cited on page 33.*

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artifical Life*, 12:229–242. *Cited on pages 69, 207, 208, and 233.*

Eric Brill. 1993. POS tagger. Software. Available from: `http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z`. *Cited on page 128.*

R. Bringhurst. 1996. *The Elements of Typographic Style*. Hartley & Marks. *Cited on page 238.*

Christoph Buchheim, Michael Jünger, and Sebastian Leipert. 2002. Improving Walker's algortihm to run in linear time. In *Proc. of the Int. Symp. on Graph Drawing (GD)*, number 2528 in LNCS, pages 344–353. Springer. *Cited on page 198.*

Gerhard N. Buurman, editor. 2005. *Total Interaction: Theory and Practice of a New Paradigm for the Design Disciplines*. Birkhäuser Basel. *Cited on page 54.*

H. Byelas and A. Telea. 2006. Visualization of areas of interest in software architecture diagrams. In *Proc. of SOFTVIS*, pages 105–114. ACM. *Cited on page 185.*

Daragh Byrne, Barry Lavelle, Gareth J. F. Johnes, and Alan F. Smeaton. 2007. Visualizing bluetooth interactions: Combining the Arc Diagram and DocuBurst techniques. In *Proc. of the BCS HCI Group Conference*, volume 2, pages 129–133. British Computer Society. *Cited on page 244.*

Lee Byron and Martin Wattenberg. 2008. Stacked graphs — geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1245–1252. *Cited on page 64.*

Stuart Card and Jock Mackinlay. 1997. The structure of the information visualization design space. In *Proc. of the IEEE Symp. on Information Visualization*, pages 92–99. IEEE Press. *Cited on page 22.*

Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, USA. *Cited on pages 14, 19, 26, 27, 28, 29, and 36.*

Stuart K. Card, George G. Robertson, and William York. 1996. The WebBook and the Web Forager: An information workspace for the World-Wide Web. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*. ACM. *Cited on page 216.*

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. In *Proc. of the Association for Computational Linguistics Workshop on Human Language Technologies (ACL-HLT)*, pages 353–361. Association for Computational Linguistics, June. *Cited on page 50.*

M. S. T. Carpendale. 1999. *A Framework for Elastic Presentation Space*. Ph.D. thesis, Simon Fraser University. *Cited on pages 29 and 30.*

M.S.T. Carpendale. 2003. Considering visual variables as a basis for information visualisation. Technical Report 2001-693-16, Department of Computer Science, University of Calgary, Calgary, Canada. *Cited on pages 14, 31, 32, and 33.*

George A. Chaikin. 1974. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3(12):346–349. *Cited on page 222.*

Chaomei Chen. 2005. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, July/August. *Cited on page 11.*

Ed Huai-hsin Chi and John Riedl. 1998. An operator interaction framework for visualization systems. In *Proc. of the IEEE Symp. on Information Visualization*, pages 63–70. *Cited on pages 29 and 30.*

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proc. of the Annual Meeting of the North American Chapter*

of the Assoc. for Computational Linguistics (NAACL), pages 26–33. *Cited on page 128.*

Jeff Clark. 2008a. Document contrast diagrams. Available from: `http://www.neoformix.com/2008/DocumentContrastDiagrams.html`. *Cited on page 144.*

Jeff Clark. 2008b. Neoformix: Discovering and illustrating patterns in data. Available from: `http://www.neoformix.com`. *Cited on page 57.*

Jeff Clark. 2009a. Twitter account graphs. Available from: `http://www.neoformix.com/2009/MoreTwitterAccountGraphs.html`. *Cited on pages 58 and 59.*

Jeff Clark. 2009b. Twitter Venn. Available from: `http://www.neoformix.com/Projects/TwitterVenn/`. *Cited on pages 57 and 58.*

Andy Clark and David J. Chalmers. 1998. The extended mind. *Analysis*, 58:10–23. *Cited on pages 14 and 22.*

William S. Cleveland and Robert McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, September. *Cited on pages 14, 27, 32, and 39.*

William S. Cleveland and Robert McGill. 1985. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833. *Cited on page 32.*

Geoffrey Cohen. 1979. Subjective probabilities and statistical parameters. In *Uncertain Outcomes*, pages 23–48. MTP Press. *Cited on page 99.*

Christopher Collins. 2004. Head-driven statistical parsing for word lattices. Master's thesis, University of Toronto. *Cited on pages 74 and 98.*

Christopher Collins and Sheelagh Carpendale. 2007. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 13(6), Nov./Dec. *Cited on page 37.*

Christopher Collins, Sheelagh Carpendale, and Gerald Penn. 2009. Docuburst: Visualizing document content using language structure. *Computer Graphics Forum (Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis))*, 28(3):1039–1046. *Cited on page 144.*

Christopher Collins and Gerald Penn. 2006. Leveraging uncertainty visualization in multilingual chatrooms. In *Proc. of Computer Supported Cooperative Work, Interactive Demonstration Session*. ACM Press, November. *Cited on page 112.*

Christopher Collins, Gerald Penn, and Sheelagh Carpendale. 2008. Interactive visualization for computational lingusitics. Tutorial at the Annual Meeting of the Association for Computational Linguistics, June. *Cited on page 43.*

Bob Coyne and Richard Sproat. 2001. Wordseye: an automatic text-to-scene conversion system. In *Proc. of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 487–496. ACM Press. *Cited on page 13.*

Brock Craft and Paul Cairns. 2005. Beyond guidelines: What can we learn from the visual information seeking mantra? In *Proc. of the Int. Conf. on Information Visualization (IV)*, pages 110–118. *Cited on page 28.*

Dennis Crouch. 2006. Forum shopping in patent cases. Available from: `http://www.patently.com/patent/2006/07/forum_shopping_.html`. *Cited on page 170.*

Catalina M. Danis, Fernanda B. Viégas, Martin Wattenberg, and Jesse Kriss. 2008. Your place or mine? Visualization as a community component. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 275–284. ACM. *Cited on page 66.*

Philip DeCamp, Amber Frid-Jimenez, Jethran Guiness, and Deb Roy. 2005. Gist Icons: Seeing meaning in large bodies of literature. In *Proc. of IEEE Symp. on Information Visualization, Poster Session*, October. *Cited on pages 76, 77, 78, 79, and 123.*

Steve DeNeefe, Kevin Knight, and Hayward H. Chan. 2005. Interactively exploring a machine translation model. In *Proc. Annual Meeting of the Assoc. for Computational Linguistics, Poster Session*. *Cited on pages 43, 49, 51, 69, 82, and 109.*

Denis Diderot, Jacques Barzun, and Ralph Henry Bowen. 1964. *Rameau's Nephew and Other Works*. Hackett Publishing, 1^{st} edition. Translation of Denis Diderot 1755. *Cited on page 9.*

John Didion. 2003. Java WordNet Library [online, cited 28 August, 2005]. Available from: `http://jwordnet.sourceforge.net`. *Cited on page 128.*

Dmitry Dimov and Brian Mulloy. 2008. Swivel preview. Website, May. Available from: http://www.swivel.com. *Cited on page 124.*

Anthony Don, Elena Zheleva, Machon Gregory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, and Catherine Plaisant. 2007. Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In *Proc. of the Conf. on Information and Knowledge Management (CIKM)*. *Cited on pages 11, 52, 53, 69, and 78.*

Judith Donath, Karrie Karahalios, and Fernanda Viégas. 1999. Visualizing conversation. In *Proc. of the Hawaii Int. Conf. on System Sciences (HICSS)*. IEEE, July. *Cited on pages 74 and 75.*

Judith Donath and Fernanda B. Viégas. 2002. The Chat Circles series: explorations in designing abstract graphical communication interfaces. In *Proc. of the Conf. on Designing Interactive Systems (DIS)*, pages 359–369. ACM Press. *Cited on page 72.*

Marian Dörk, Sheelagh Carpendale, Christopher Collins, and Carey Williamson. 2008. VisGets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1205–1213, Nov./Dec. *Cited on pages 83, 124, 142, and 145.*

Johanna Drucker. 1984. Letterpress language: Typography as a medium for the visual representations of language. *Leonardo*, 17(1):8–16. *Cited on page 24.*

Paul R. Dunn. 2007. Visuwords online graphical dictionary [online, cited 27 April, 2007]. Available from: http://visuwords.com. *Cited on page 214.*

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74. *Cited on pages 159 and 250.*

Tim Dwyer, Kim Marriott, Falk Schreiber, Peter J. Stuckley, Michael Woodward, and Michael Wybrow. 2008. Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1293–1300, Nov./Dec. *Cited on pages 180 and 184.*

Stephen G. Eick. 1994. Graphically displaying text. *Journal of Computational Graphics and Statistics*, 3(2). *Cited on page 74.*

N. Elmqvist, P. Dragicevic, and J.-D. Fekete. 2008. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1141–1148. *Cited on page 180.*

Thomas Erickson, Wendy A. Kellogg, Mark Laff, Jeremy Sussman, Tracee Vetting Wolf, Christina A. Halverson, and Denise Edwards. 2006. A persistent chat space for work groups: The design, evaluation and deployment of loops. In *Proc. of the Conf. on Designing Interactive Systems (DIS)*, June. *Cited on pages 70 and 72.*

Thomas Erickson and Mark R. Laff. 2001. The design of the 'Babble' timeline: a social proxy for visualizing group activity over time. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 329–330. ACM. *Cited on pages 70 and 72.*

Matt Ericson. 2007. Visualizing data for the masses: Information graphics at The New York Times. Keynote Address at IEEE InfoVis. Available from: `http://ericson.net/infovis/`. *Cited on pages 54 and 66.*

Andrea Falletto, Paolo Prinetto, and Gabriele Tiotto. 2009. *Electronic Healthcare*, volume 1867 of *LNCS*, chapter An Avatar-Based Italian Sign Language Visualization System, pages 154–160. Springer. *Cited on page 44.*

Jonathan Feinberg. 2008. Wordle: Beautiful word clouds [online, cited 2 December, 2008]. Available from: `http://www.wordle.net`. *Cited on pages 11, 68, and 158.*

Jean-Daniel Fekete and Nicole Dufournaud. 2000. Compus visualization and analysis of structured documents for understanding social life in the 16th century. In *Proc. of the Joint Conf. on Digital Libraries (JCDL)*. ACM. *Cited on page 78.*

Jean-Daniel Fekete, David Wang, Niem Dang, Aleks Aris, and Catherine Plaisant. 2003. Overlaying graph links on Treemaps. In *Proc. of IEEE Symp. on Information Visualization, Poster Session*, pages 82–83. *Cited on pages 184, 207, and 210.*

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA. *Cited on pages 6, 123, and 126.*

Alex Franz and Thorsten Brants. 2006. All our N-gram are belong to you [online, cited 19 August, 2009]. Available from: `http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html`. *Cited on page 10.*

Ralph Freese. 2004. *Concept Lattices*, volume 2961 of *Lecture Notes in Computer Science*, chapter Automated Lattice Drawing, pages 112–127. Springer-Verlag. *Cited on pages 101 and 104.*

Wolfgang Freiler, Kresšimir Matković, and Helwig Hauser. 2008. Interactive visual analytics of set-typed data. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1340–1347, Nov./Dec. *Cited on page 180.*

Ben Fry. 2007. zipdecode. http://acg.media.mit.edu/people/fry/zipdecode/. *Cited on page 228.*

Ben Fry and Casey Reas. 2008. Processing. Website and software, May. Available from: `http://www.processing.org`. *Cited on page 57.*

George W. Furnas. 1986. Generalized fisheye views. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 16–23. ACM Press, April. *Cited on page 132.*

Government Printing Office. 2008. Congressional directory [online, cited 10 August, 2009]. Available from: `http://www.gpoaccess.gov/cdirectory`. *Cited on page 228.*

Saul Greenberg and Bill Buxton. 2008. Usability evaluation considered harmful (some of the time). In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 111–120. *Cited on page 40.*

Michelle L. Gregory, Nancy Chinchor, Paul Whitney, Richard Carter, Elizabeth Hetzler, and Alan Turner. 2006. User-directed sentiment analysis: Visualizing the affective content of documents. In *Proc. of the Workshop on Sentiment and Subjectivity in Text*, pages 23–30. ACL. *Cited on pages 11, 74, and 142.*

Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. 2007. Exploring and reducing the effects of orientation on text readability in volumetric displays. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 483–492. *Cited on pages 51 and 238.*

Yves Guiard, Michel Beaudouin-Lafon, Yangzhou Du, Caroline Appert, Jean-Daniel Fekete, and Olivier Chapuis. 2006. Shakespeare's complete works as a benchmark for evaluating multiscale document navigation techniques. In *Proc. of the Conference on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. *Cited on page 49.*

Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: Clutter-free visualization of off-screen locations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 787–796. ACM. *Cited on page 266.*

Keith B. Hall. 2005. *Best-first Word-lattice Parsing: Techniques for integrated syntactic language modeling*. Ph.D. thesis, Brown University, Providence, USA. *Cited on page 98.*

Mark Hancock, Otmar Hilliges, Christopher Collins, Diminikus Baur, and Sheelagh Carpendale. 2009. Exploring tangible and direct touch interfaces for manipulating 2d and 3d information on a digital table. In *Proc. of the Int. Conf. on Interactive Tabletops and Surfaces*. ACM. *Cited on page 244.*

Jeffrey T. Hancock, Christopher Landrigan, and Courtney Silver. 2007. Expressing emotion in text-based communication. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 929–932. ACM. *Cited on page 7.*

Jonathan Harris and Sepandar Kamvar. 2006. We feel fine. Available from: `http://www.wefeelfine.org/` [online, cited 12 August, 2006]. *Cited on pages 11, 59, 60, and 66.*

Yusef Hassan-Montero and Víctor Herrero-Solana. 2006. Improving tag-clouds as visual information retrieval interfaces. In *Proc. of the Int. Conf. on Multidisciplary Information Sciences and Technologies*. *Cited on pages 124 and 125.*

E. Havelock. 1982. *The Literate Revolution in Greece and its Cultural Consequences*. Princeton University Press. *Cited on page 7.*

Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. 2002. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8, January. *Cited on pages 11, 12, 62, 63, 81, and 144.*

Christopher G. Healey. 2007. Perception in visualization. Website. Available from: http://www.csc.ncsu.edu/faculty/healey/PP. *Cited on pages 35 and 144.*

Christopher G. Healey, Kellogg S. Booth, and James T. Enns. 1996. High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction*, 3(2):107–135. *Cited on pages 27, 33, and 34.*

Marti A. Hearst. 1995. Tilebars: Visualization of term distribution information in full text information access. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 59–66. ACM Press. *Cited on pages 11, 78, and 83.*

Marti A. Hearst. 2002. Informational visualization and presentation. PowerPoint Slides, March. Available from: http://www.sims.berkeley.edu/courses/is247/s02/lectures/TextAndSearch.ppt. *Cited on pages 4 and 82.*

Marti A. Hearst. 2006. Design recommentations for hierarchical faceted search interfaces. In Andrei Z. Broder and Yoelle S. Maarek, editors, *Proc. SIGIR Workshop on Faceted Search*, pages 26–30. *Cited on page 62.*

Marti A. Hearst. 2008. What's up with tag clouds? *Perceptual Edge Newsletter*, May. *Cited on page 124.*

Marti Hearst. 2009. *Search User Interfaces*. Cambridge University Press. *Cited on pages 11 and 82.*

Marti A. Hearst and Chandu Karadi. 1997. Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proc. of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–255. ACM Press. *Cited on page 158.*

Jeffrey Heer. 2007. congress. http://www.prefuse.org/gallery/congress. *Cited on page 228.*

Jeffrey Michael Heer. 2008. *Supporting Asynchronous Collaboration for Interactive Visualziation*. Ph.D. thesis, Berkeley. *Cited on page 70.*

Jeffrey Heer, Stuart K. Card, and James A. Landay. 2005. prefuse: a toolkit for interactive information visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*. ACM Press, April. *Cited on pages 127, 226, and 228.*

Jeffrey Heer and danah boyd. 2005. Vizster: Visualizing online social networks. In *Proc. of the IEEE Symp. on Information Visualization*. *Cited on pages 180 and 183.*

Mary Hegarty. 2004. Diagrams in the mind and in the world: Relations between internal and external visualizations. In *Diagrammatic Representation and Inference*, volume 2980/2004 of *Lecture Notes in Computer Science*, pages 121–132. Springer. *Cited on page 19.*

Christian Heine and Gerik Scheuermann. 2007. Manual clustering refinement using interaction with blobs. In *Proc. of Eurographics/IEEE-VGTC Symp. on Visualization*. The Eurographics Association. *Cited on page 185.*

Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. 2007. Nodetrix. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 13(6), Nov./Dec. *Cited on page 28.*

Susan C. Herring, John C. Paolillo, Irene Ramos-Vielba, Inna Kouper, Elijah Write, Sharon Stoerger, Lois Ann Scheidt, and Benjamin Clark. 2007. Language networks on LiveJournal. In *Proc. of the Hawaii Int. Conf. on System Sciences (HICSS)*. IEEE. *Cited on page 109.*

Beth Hetzler, W. Michelle Harris, Susan Havre, and Paul Whitney. 1998a. Visualizing the full spectrum of document relationships. In *Structures and Relations in Knowledge Organization, Proc. of the 5th Conference of the International Society for Knowledge Organization*, pages 168–175. ERGON-Verlag. *Cited on page 11.*

Beth Hetzler, Paul Whitney, Lou Martucci, and Jim Thomas. 1998b. Multi-faceted insight through interoperable visual information analysis paradigms. In *Proc. of the IEEE Symp. on Information Visualization*, pages 137–144, October. *Cited on page 76.*

William C. Hill, James D. Hollan, Dave Wroblewski, and Tim McCandless. 1992. Edit wear and read wear. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI). Cited on pages 49 and 52.*

Uta Hinrichs, Holly Schmidt, and Sheelagh Carpendale. 2008. EMDialog: Bringing information visualization into the museum. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1181–1188. *Cited on page 66.*

Julia Hirschberg and Christine Nakatani. 1998. Acoustic indicators of topic segmentation. In Robert H. Mannell and Jordi Robert-Ribes, editors, *Proc. of the Int. Conf. on Spoken Language Processing*, volume 4, pages 1255–1258, Sydney, Australia, November. Australian Speech Science and Technology Association, Incorporated (ASSTA). *Cited on page 241.*

Danny Holten. 2006. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Symp. on Information Visualization)*, 12(5):741–748, Sept.–Oct. *Cited on pages 207, 210, 211, and 230.*

Danny Holten and Jarke J. van Wijk. 2009. Force-directed edge bundling for graph visualization. *Computer Graphics Forum (Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis))*, 28(3). *Cited on page 230.*

Robert E. Horn. 1999. *Visual Language: Global Communication for the 21st Century*. MacroVU Press, Bainbridge Island, USA. *Cited on page 7.*

E. Hutchins. 1995. *Cognition in the Wild*. MIT Press. *Cited on page 22.*

IBM Research. 2009. Many Eyes comparison tag cloud [online, cited 25 March, 2009]. Available from: `http://manyeyes.alphaworks.ibm.com/manyeyes/page/Tag_Cloud.html`. *Cited on page 144.*

Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2006. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 625–632. ACL. *Cited on page 48.*

Diana Z. Inkpen and Graeme Hirst. 2002. Acquiring collocations for lexical choice between near-synonyms. In *Unsupervised Lexical Acquisition: Proc. of ACL SIGLEX*, pages 67–76. *Cited on page 159.*

Alfred Inselberg. 1985. The plane with parallel coordinates. *Visual Computer*, 1(4):69–91. *Cited on page 16.*

Alfred Inselberg and Bernard Dimsdale. 1990. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. of IEEE Visualization*, pages 361–378. *Cited on pages 142, 143, 222, and 251.*

Petra Isenberg and Danyel Fisher. 2009. Collaborative brusing and linking for co-located visual analytics of document collections. *Computer Graphics Forum (Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis))*, 28(3):1031–1038, June. *Cited on page 70.*

Petra Isenberg, Anthony Tang, and Sheelagh Carpendale. 2008. An exploratory study of visual information analysis. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*. ACM. *Cited on pages 29 and 70.*

Laura A. Janusik and Andrew D. Wolvin. 2009. 24 hours in a day: A listening update to the time studies. *International Journal of Listening*, 23(2):104–120. *Cited on pages 3 and 9.*

Frederick Jelinek. 1998. *Statistical Methods for Speech Recognition*. MIT Press. *Cited on page 96.*

Eric Joanis. 2002. Automatic verb classification using a general feature space. Master's thesis, University of Toronto. *Cited on page 82.*

Chris R. Johnson and Allan R. Sanderson. 2003. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23:6–10, September. *Cited on page 100.*

Brian Johnson and Ben Shneiderman. 1991. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proc. of IEEE Visualization*, pages 284–291. IEEE Computer Society. *Cited on page 228.*

Steve Jones and S. McInnes. 1998. Graphical query specification and dynamic result previews for a digital library. In *Proc. of the ACM Symp. on User Interface Software and Technology (UIST)*, pages 143–151, November. *Cited on page 82.*

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing*. Prentice Hall, New Jersey, USA, 1 edition. *Cited on page 97.*

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, 2 edition. *Cited on page 45.*

Jaap Kamps and Maarten Marx. 2002a. Visualizing WordNet structure. In *Proc. of the 1st International Conference on Global WordNet*, pages 182–186. *Cited on pages 127 and 214.*

Jaap Kamps and Maarten Marx. 2002b. Words with attitude. In *Proc. of the 1st International Conference on Global WordNet*, pages 332–341. *Cited on page 127.*

Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proc. of the 4th Annual Conference on Language Resources and Evaluation (LREC)*, pages 1115–1118. *Cited on page 127.*

Thomas Kemp and Thomas Schaaf. 1997. Estimating confidence using word lattices. In *Proc. of Eurospeech*, pages 827–830. *Cited on page 114.*

Sebastian Kempken, Thomas Pilz, and Wolfram Luther. 2007. Visualization of rule productivity in deriving non-standard spellings. In *Proc. of SPIE-IS&T Electronic Imaging (VDA '07)*, volume 6495. *Cited on pages 11 and 69.*

Kenneth E. Kidd. 1965. Birch-bark scrolls in archaeological contexts. *American Antiquity*, 4:480–483. *Cited on page 8.*

Kenneth E. Kidd. 1981. A radiocarbon date on a Midewiwin scroll from Burntside Lake, Ontario. *Ontario Archaeology*, 35:41–43. *Cited on page 8.*

A. Kilgariff and T. Rose. 1998. Measures for corpus similarity and homogeneity. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–52. *Cited on pages 158 and 159.*

D. Brett King and Michael Wertheimer. 2005. *Max Wertheimer and Gestalt Theory*. Transaction, London, UK. *Cited on page 36.*

Matthew G. Kirschenbaum. 2004. "So the colors cover the wires": Interface, aesthetics, and usability. In Susan Schreibman, Ray Siemens, and John Unsworth, editors, *A Companion to Digital Humanities*, chapter 34. Blackwell. *Cited on page 12.*

David Kirsh and Paul Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18:513–549. *Cited on page 26.*

Bradley Kjell, W. Addison Woods, and Ophir Frieder. 1994. Discrimination of authorship using visualization. *Information Processing & Management*, 30(1):141–150. *Cited on page 74.*

Torkel Klingberg. 2008. *The Overflowing Brain: Information Overload and the Limits of Working Memory*. Oxford University Press. *Cited on page 4.*

Kevin Knight. 1999. A statistical MT tutorial workbook, April. *Cited on page 96.*

Philipp Koehn. 2003. European Parliament proceedings parallel corpus, 1996–2003. Available from: http://people.csail.mit.edu/~koehn/publications/europarl.ps. *Cited on page 110.*

Philipp Koehn. 2004. *Pharaoh User Manual and Description for Version 1.2*. USC ISI, August. Available from: http://www.isi.edu/licensed-sw/pharaoh/manual-v1.2.ps. *Cited on pages 96 and 110.*

Robert Kosara, Christopher G. Healey, Victoria Interrante, David H. Laidlaw, and Colin Ware. 2003. Thoughts on user studies: Why, how, and when. *IEEE Computer Graphics and Applications*, 23(4):20–25. (Visualization Viewpoints). *Cited on page 39.*

Robert Kosara, Silvia Miksch, and Helwig Hauser. 2001. Semantic depth of field. In *Proc. of the IEEE Symp. on Information Visualization*. *Cited on pages 107 and 238.*

T. K. Landauer. 1986. How much do people remember? Some estimates of the quantity of learned information in long-term memory. *Cognitive Science*, 10(4):477–493. *Cited on page 19.*

Clarence Larkin. 1918. *Dispensational Truth*. PreservedWords.com. Accessed online http://www.preservedwords.com/charts.htm. *Cited on pages 24 and 25.*

Jill H. Larkin and Herbert A. Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–99. *Cited on pages 24, 26, and 37.*

K. Larson, M. van Dantzich, M. Czerwinski, and G. Robertson. 2000. Text in 3D: Some legibility results. In *ACM CHI Extended Abstracts*, pages 145–146. *Cited on page 238.*

George Legrady. 2005. Making visible the invisible: Seattle library data flow visualization. In *Proc. of Digital Culture and Heritage (ICHIM)*. *Cited on page 66.*

Anton Leuski, Chin-Yew Lin, and Eduard Hovy. 2003. iNeATS: Interactive mult-document summarization. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Interactive Posters and Demos Session, July. *Cited on pages 49 and 50.*

Golan Levin and Zachary Lieberman. 2004. In-situ speech visualization in real-time interactive installation and performance. In *Proc. of the*

Int. Symp. Non-photorealistic Animation and Rendering, pages 7–14. ACM. *Cited on pages 44 and 66.*

Golan Levin, Kamal Nigam, and Johnathan Feinberg. 2005. The Dumpster: A portrait of romantic breakups. Available from: `http://artport.whitney.org/commissions/thedumpster/`. *Cited on page 59.*

J. C. R. Licklider. 1960. Man–computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11. *Cited on page 27.*

John Light and J.D. Miller. 2002. Miramar: A 3D workplace. In *Proc. of IEEE Intl. Professional Communication Conf.*, pages 271–282. *Cited on pages 216 and 217.*

Chin-Yew Lin and Eduard Hovy. 2002. From single to multi-document summarization: a prototype system and its evaluation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 457–464. *Cited on page 143.*

Zhicheng Liu, Nancy J. Nersessian, and John T. Stasko. 2008. Distributed cognition as a theoretical framework for information visualization. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1173–1180. *Cited on page 138.*

Peng Liu and Frank K. Soong. 2006. Word graph based speech recognition error correction by handwriting input. In *Proc. of the Int. Conf. on Multimodal Interfaces*, pages 339–346, November. *Cited on page 108.*

W. E. Lorensen and H. E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. of the Int. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 163–169. *Cited on pages 192 and 196.*

Hans Peter Luhn. 1960. Keyword-in-context index for technical literature. *American Documentation*, 11(4):288–295. *Cited on pages 43, 52, 135, and 156.*

Fritz Machlup. 1963. *The Production and Distribution of Knowledge in the United States*. Princeton University Press. *Cited on page 20.*

Carl Malamud. 2008. US Federal Reporter 2^{nd} and 3^{rd} ed., bulk download. Available from: `http://bulk.resource.org/`. *Cited on page 142.*

Christopher D. Manning, Kevin Jansz, and Nitin Indurkhya. 2001. Kirrkirr: Software for browsing and visual exploration of a structured Walpiri dictionary. *Literary and Linguistic Computing*, 16(2):135–151. *Cited on pages 46 and 214.*

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press. *Cited on page 45.*

Kent Manske. 2000. Writing forms [online, cited 1 April, 2005]. Available from: `http://www.foothill.edu/~kmanske/Lecture_notes/PDF/Writing_Forms.pdf`. *Cited on page 7.*

D. Marr. 1982. *A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company. *Cited on page 23.*

Michael J. McGuffin, Liviu Tancau, and Ravin Balakrishnan. 2003. Using deformations for browsing volumetric data. In *Proc. of IEEE Visualization*, pages 401–408, October. *Cited on page 217.*

Sean M. McNee and Ben Arnette. 2008. Productivity as a metric for visual analytics: reflections on e-discovery. In *Proc. of the Conference on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM. *Cited on page 39.*

Wolfgang Mieder. 2004. *Proverbs: A Handbook*. Greenwood Publishing Group. *Cited on page 3.*

R. Mihalcea and P. Tarau. 2004. Textrank: Bringing order into texts. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 404–411. ACL. *Cited on pages 123 and 125.*

J. G. Miller. 1960. Information input overload and psychopathology. *American Journal of Psychiatry*, 116:695–704. *Cited on page 3.*

J. G. Miller. 1962. Information input overload. In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, editors, *Proc. of Self-Organizing Systems*, pages 61–78. *Cited on page 3.*

J. G. Miller. 1964. Coping with administrators' information overload. In *Report of the First Institute on Medical School Administration*, pages 47–54, Evanston. Association of American Medical Colleges. *Cited on page 3.*

George A. Miller, Christiane Fellbaum, Randee Tengi, Susanne Wolff, Pamela Wakefield, Helen Langone, and Benjamin Haskell. 2007. WordNet: A lexical database for the English language, March. Available from: http://www.cogsci.princeton.edu/cgi-bin/webwn [online, cited 31 March, 2007]. *Cited on pages 88, 207, and 214.*

Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. 1995. Layout adjustment and the mental map. *J. Visual Languages and Computing*, 6:183–210. *Cited on page 251.*

Andrew Vande Moere. 2009. Information aesthetics weblog. Available from: http://infosthetics.com. *Cited on page 10.*

Robert C. Moore. 2004. On log-likelihood-ratios and the significance of rare events. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*. ACL. *Cited on page 160.*

Franco Moretti. 2005. *Graphs, Maps, Trees*. Verso. *Cited on pages 43, 51, and 141.*

Meredith Ringel Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. 2004. Beyond "social protocols": Multi-user coordination policies for co-located groupware. In *Proc. of Computer-Supported Cooperative Work*. *Cited on page 159.*

Martin Mueller. 2008. Comparing word form counts (WordHoard documentation) [online, cited 20 August, 2008]. Available from: http://wordhoard.northwestern.edu/userman/analysis-comparewords.html. *Cited on pages 11, 54, and 160.*

Tamara Munzner. 2000. *Interactive Visualizations of Large Graphs and Networks*. Ph.D. thesis, Stanford University. *Cited on page 82.*

Tamara Munzner, François Guimbretière, and George Robertson. 1999. Constellation: A visualizaiton tool for linguistic queries from MindNet. In *Proc. of the IEEE Symp. on Information Visualization*. IEEE. *Cited on page 70.*

Tamara Munzner, Francois Guimbretière, Serdar Tasiran, Li Zhang, and Yunhong Zhou. 2003. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453–462. SIGGRAPH 2003. *Cited on pages 126 and 129.*

Netcraft. 2009. April 2009 web server survey [online, cited 19 August, 2009]. Available from: http://news.netcraft.com/archives/2009/04/06/april_2009_web_server_survey.html. *Cited on page 10.*

Hansjeorg Neth and Stephen J. Payne. 2002. Thinking by doing? Epistemic actions in the Tower of Hanoi. In *Proc. of the Annual Conf. of the Cognitive Science Society*, pages 691–696. *Cited on page 26.*

Petra Neumann, Stefan Schlechtweg, and M. S. T. Carpendale. 2005. Arc-Trees: Visualizing relation in hierarchical data. In K. W. Brodlie, D. J. Duke, and K. I. Joy, editors, *Proc. of Eurographics/IEEE-VGTC Symp. on Visualization*, pages 53–60. The Eurographics Association. *Cited on pages 207 and 210.*

New York Times. 2009. New York Times developer network. Available from: http://developer.nytimes.com/. *Cited on page 10.*

Donald A. Norman. 1993. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publishing Co., Boston, USA. *Cited on pages 21 and 22.*

Donald A. Norman. 2002. Emotion & design: Attractive things work better. *Interactions*, 9(4):36–42. *Cited on page 54.*

Chris North and Ben Shneiderman. 2000. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proc. of Advanced Visual Interfaces*, pages 128–135, May. *Cited on page 210.*

Daniela Oelke, Peter Bak, Daniel A. Keim, Mark Last, and Guy Danon. 2008. Visual evaluation of text features for document summarization and analysis. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 75–82. *Cited on page 78.*

David R. Olson. 1996a. Towards a psychology of literacy: On the relationships between speech and writing. *Cognition*, 60:83–104. *Cited on page 7.*

David R. Olson. 1996b. *The world on paper*. Cambridge University Press. *Cited on page 8.*

Marian Olteanu. 2006. Phramer: An open-source statistical phrase-based MT decoder. Software, May. Available from: http://www.phramer.org. *Cited on page 110.*

Open Library. 2009. Open library API. Available from: `http://openlibrary.org/dev/docs`. *Cited on page 10.*

W. Bradford Paley. 2002. TextArc: Showing word frequency and distribution in text. In *Proc. of the IEEE Symp. on Information Visualization*, Poster. IEEE Computer Society, October. *Cited on pages 11, 76, 77, 89, and 123.*

Han Pan and Chao Wang. 2009. Visualization of text duplicates in documents. Master's thesis, Växjö University. *Cited on page 244.*

Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. Technical Report UMSI 2005/25, University of Minnesota Supercomputing Institute. *Cited on page 216.*

Adam Perer and Ben Shneiderman. 2006. Balancing systematic and flexible exploration of social networks. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Symp. on Information Visualization)*, 12(5):693–700. *Cited on page 180.*

Henry Petroski. 2000. *The Book on the Bookshelf*. Vintage. *Cited on page 121.*

D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. 2005. Flow map layout. In *Proc. of the IEEE Symp. on Information Visualization*, pages 219–224. *Cited on pages 184 and 191.*

Steven Pinker. 1997. *How the Mind Works*. W. W. Norton & Co. *Cited on pages 3 and 35.*

Catherine Plaisant. 2004. The challenge of information visualization evaluation. In *Proc. of the ACM Conf. on Advanced Visual Interfaces (AVI)*, pages 109–116. ACM Press. *Cited on page 39.*

Catherine Plaisant, James Rose, Bei Yu, Loretta Auvil, Matthew G. Kirschenbaum, Martha Nell Smith, Tanya Clement, and Greg Lord. 2006. Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In *Proc. of the Joint Conf. on Digital Libraries (JCDL)*. *Cited on pages 11, 21, and 74.*

Sabine Ploux and Hyungsuk Ji. 2003. A model for matching semantic maps between languages (French/English, English/French). *Computational Linguistics*, 29(2):155–178, June. *Cited on pages 46, 48, 89, and 214.*

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137. *Cited on page 164.*

Stefanie Posavec. 2008. Literary organism. In *IEEE InfoVis Art Exhibition*. Available from: http://www.itsbeenreal.co.uk/index.php?/wwwords/literary-organism/. *Cited on pages 61 and 62.*

Zachary Pousman, John T. Stasko, and Machael Mateas. 2007. Casual information visualization: Depictions of data in everyday life. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 13(6):1145–1152. *Cited on pages xxiii and 57.*

Paul Rayson and Roger Garside. 2000. Comparing corpora using frequency profiling. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL), Workshop on Comparing Corpora*, pages 1–6. *Cited on pages 158 and 159.*

Magnus Rembold and Jürgen Späth. 2006. Graphical visualization of text similarities in essays in a book [online, cited 10 August, 2006]. Available from: http://www.munterbund.de/visualisierung_textaehnlichkeiten/essay.html. *Cited on pages 54, 56, and 144.*

P. Rheingans and S. Joshi. 1999. Visualization of molecules with positional uncertainty. In *Proc. of Data Visualization*, pages 299–306. Springer-Verlag. *Cited on page 101.*

George G. Robertson and Jock D. Mackinlay. 1993. The document lens. In *Proc. of the ACM Symp. on User Interface Software and Technology (UIST)*, pages 101–108. ACM. *Cited on pages 49 and 52.*

Randall M. Rohrer, John L. Sibert, and David S. Ebert. 1999. A shape-based visual interface for text retrieval. *IEEE Computer Graphics and Applications*, 19(5):40–47, September. *Cited on page 12.*

Hans Rosling. 2009. Gapminder [online, cited 31 March, 2009]. Available from: http://www.gapminder.org/. *Cited on page 203.*

Pruvi Saraiya, Chris North, and Karen Duca. 2005. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456. *Cited on page 39.*

Mike Scaife and Yvonne Rogers. 1996. External cognition: How do graphical representations work? *International Journal of Human-Computer Studies*, 45(2):185–213. *Cited on pages 22 and 26.*

Susan Schreibman, Ray Siemens, and John Unsworth, editors. 2004. *A Companion to Digital Humanities*. Blackwell. Accessed online at http://www.digitalhumanities.org/companion/. *Cited on pages 11, 43, and 52.*

Stacey D. Scott, Neal Lesh, and Gunnar W. Klau. 2002. Investigating human-computer optimization. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 155–162, April. Available from: `http://scripts.mit.edu/~sdscott/wiki/uploads/Main/scott_chi2002.pdf`. *Cited on pages 26 and 103.*

John Searle. 1980. Minds, brains and programs. *Behavioral and Brain Sciences*, 3(3):417–457. *Cited on page 4.*

Fred R. Shapiro. 2003. The politically correct U.S. Supreme Court and the motherfucking Texas Court of Criminal Appeals: Using legal databases to trace the origins of words. In *Language and the Law: Proceedings of a Conference*, pages 367–372. William S. Hein & Co. *Cited on page 145.*

Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symp. on Visual Languages*, pages 336–343. IEEE Press. *Cited on pages 14, 27, 28, and 38.*

Ben Shneiderman and Aleks Aris. 2006. Network visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Symp. on Information Visualization)*, 12(5):733–740, Sept.–Oct. *Cited on pages 144 and 211.*

Ben Shneiderman and Catherine Plaisant. 2006. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proc. of the Conference on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. *Cited on page 39.*

Herbert A. Simon and John R. Hayes. 1976. The understanding process: Problem isomorphs. *Cognitive Psychology*, 8:165–190. *Cited on page 23.*

Stefan Sinclair. 2006. Hyperpo: Digital text reading environment. Available from: `http://hyperpo.org`. *Cited on page 11.*

Stefan Sinclair and Gregory Rockwell. 2009. Voyeur tools: Reveal your texts. Available from: `http://hermeneuti.ca/node/23`. *Cited on pages 11, 52, and 53.*

Marc A. Smith and Andrew T. Fiore. 2001. Visualization components for persistent conversations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 136–143. ACM Press. *Cited on page 74.*

Noah A. Smith and Michael E. Jahr. 2000. Cairo: An alignment visualization tool. In *Proc. of the Int. Conf. on Language Resources and Evaluation*, pages 549–552. *Cited on pages 43, 49, 82, and 109.*

Bridget Somekh and Cathy Lewin, editors. 2004. *Research Methods in the Social Sciences*. Sage Publications, London, 3rd edition. *Cited on page 176.*

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21. *Cited on page 159.*

Richard Sproat. 2005. What is computational linguistics? [online, cited 28 August, 2009]. Available from: `http://www.aclweb.org/archive/misc/what.html`. *Cited on page 45.*

Stamen Design. 2006. Backchannel. Available from: `http://stamen.com/projects/backchannel`. *Cited on pages 72 and 73.*

John Stasko, Carsten Görg, Zhicheng Liu, and Kanupriya Singhal. 2007. Jigsaw: Supporting investigative analysis through interactive visualization. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)*, pages 131–138. *Cited on pages 11, 64, 65, 74, and 144.*

John Stasko and Eugene Zhang. 2000. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proc. of the IEEE Symp. on Information Visualization*, pages 57–65. *Cited on page 126.*

Graham A. Stephen. 1994. *String Searching Algorithms*. World Scientific. *Cited on page 66.*

Andreas Stolcke. 2002. SRILM — an extensible language modeling toolkit. In *Proc. of the Int. Conf. on Spoken Language Processing*, volume 2, pages 901–904. *Cited on pages 102 and 110.*

Maureen C. Stone. 2003. *A Field Guide to Digital Color*. AK Peters, Ltd. *Cited on pages 14, 27, 33, 122, 129, and 238.*

Thomas Strothotte, Maic Masuch, and Tobias Isenberg. 1999. Visualizing knowledge about virtual reconstructions of ancient architecture. In *Proc. of Computer Graphics International*, pages 36–43. *Cited on page 101.*

Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACL Transactions on Computer-Human Interaction*, 8(1):60–98. *Cited on page 112.*

Sunlight Foundation. 2009. Sunlight labs API. Available from: `http://www.sunlightlabs.com/`. *Cited on page 10.*

Carl Tashian. 2009. Lost in translation. Available from: `http://tashian.com/multibabel/`. *Cited on page 175.*

Annie Tat. 2007. Visualizing digital communication. Master's thesis, University of Calgary. *Cited on pages 200, 201, and 203.*

A. Tat and M. S. T. Carpendale. 2002. Visualising human dialog. In *Proc. of the Int. Conf. on Information Visualization (IV)*, pages 16–21. *Cited on pages 70, 72, 73, and 109.*

Annie Tat and Sheelagh Carpendale. 2006. CrystalChat: Visualizing personal chat history. In *Proc. of the Hawaii Int. Conf. on System Sciences (HICSS). Cited on pages 72 and 109.*

Nina Thiessen. 2004. Connection maps: A new way to visualize similarity relationships. Master's thesis, University of Toronto. *Cited on pages 82 and 84.*

ThinkMap. 2005. ThinkMap visual thesaurus, April. Available from: `http://www.visualthesaurus.com` [online, cited 10 April, 2005]. *Cited on page 127.*

James J. Thomas and Kristin A. Cook. 2005. *Illuminating the Path*. IEEE Press. *Cited on page 11.*

Peter Meijes Tiersma. 1999. *Legal Language*. University of Chicago Press. *Cited on pages 141 and 145.*

Alvin Toffler. 1971. *Future Shock*. Bantam, 3rd edition. *Cited on page 3.*

A. Treisman and S. Gormican. 1988. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, 96:15–48. *Cited on page 35.*

Edward R. Tufte. 1990. *Envisioning Information*. Graphics Press, Cheshire, USA. *Cited on pages 26, 36, and 52.*

Edward R. Tufte. 2001. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, USA, 2^{nd} edition. *Cited on pages 14, 24, 27, 36, 101, and 104.*

Edward R. Tufte. 2006. *Beautiful Evidence*. Graphics Press. *Cited on pages 27, 36, and 52.*

Barbara Tversky. 2004. *The Cambridge Handbook of Thinking and Reasoning*, chapter Visuospatial Reasoning. Cambridge University Press. *Cited on pages 4, 14, and 24.*

Twitter. 2009. Twitter API wiki. Available from: `http://apiwiki.twitter.com/`. *Cited on page 10.*

Saša Šantić. 2006. Lascaux painting. Wikimedia Commons Image, February. Creative Commons Attribution ShareAlike 3.0 License. Available from: `http://en.wikipedia.org/wiki/File:Lascaux_painting.jpg`. *Cited on page 8.*

P. Valtchev, D. Grosser, C. Roume, and M. Rouane Hacene. 2003. Galicia: an open platform for lattices. In Aldo de Moor, Wilfried Lex, and Bernhard Ganter, editors, *Contributions to the 11th Conference on Conceptual Structures*, pages 241–254. Verlag Shaker. *Cited on pages 99 and 101.*

Frank van Ham, Martin Wattenberg, and Fernanda B. Viégas. 2009. Mapping text with Phrase Nets. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 15(6):1169–1176, Nov./Dec. *Cited on page 66.*

Martijn van Welie, Gerrit C. van der Veer, and Anton Eliëns. 2000. Patterns as tools for user interface design. In *Proc. Int. Workshop on Tools for Working with Guidelines*, pages 313–324. *Cited on page 29.*

Fernanda Viégas and Judith Donath. 1999. Chat Circles. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 9–16. *Cited on pages 70 and 109.*

Fernanda B. Viégas, Scott Golder, and Judith Donath. 2006. Visualizing email content: Portraying relationships from conversational histories. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*. *Cited on pages 80, 81, 144, 159, and 161.*

Fernanda B. Viégas and Marc Smith. 2004. Newsgroup Crowds and AuthorLines: Visualizing the activity of individuals in conversational cyberspaces. In *Proc. of the Hawaii Int. Conf. on System Sciences (HICSS)*. IEEE Computer Society. *Cited on pages 11 and 74.*

Fernanda Viégas and Martin Wattenberg. 2007. Artistic data visualization: Beyond visual analytics. In *Proc. of HCI Int.*, volume 4564 of *LNCS*, pages 182–191. Springer. *Cited on page 59.*

Fernanda B. Viégas and Martin Wattenberg. 2008. Tag clouds and the case for vernacular visualization. *Interactions*, 15(4):49–52. *Cited on pages xxiv and 57.*

Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. 2004. Studying cooperation and conflict between authors with History Flow visualizations. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 575–582. ACM Press. *Cited on pages 11, 74, and 145.*

Fernanda B. Viégas, Martin Wattenberg, and Jonathan Feinberg. 2009. Participatory visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 15(6):1137–1144, Nov./Dec. *Cited on pages 12, 66, 68, 121, and 124.*

Fernanda B. Viégas, Martin Wattenberg, Jesse Kriss, and Matt McKeon. 2007. Many Eyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 13(6):1121–1128, Nov./Dec. *Cited on pages 37, 66, 121, and 156.*

Colin Ware. 2004. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd edition. *Cited on pages 14, 19, 27, 34, 35, 36, 101, 105, and 129.*

McKenzie Wark. 1997. *The Virtual Republic*. Allen and Unwin, St. Leonards. *Cited on page 20.*

Mark Warscchauer. 2004. *Technology and Social Inclusion: Rethinking the Digital Divide*. MIT Press. *Cited on page 20.*

Nayuko Watanabe, Motoi Washida, and Takeo Igarashi. 2007. Bubble Clusters: An interface for manipulating spatial aggregation of graphical objects. In *Proc. of the ACM Symp. on User Interface Software and Technology (UIST)*. ACM, October. *Cited on pages 182 and 185.*

Margaux Watt. 2008. Up to speed: Canada reads. CBC Radio Interview, February. *Cited on page 138.*

Martin Wattenberg. 2002. Arc Diagrams: Visualizing structure in strings. In *Proc. of the IEEE Symp. on Information Visualization*. *Cited on pages 74 and 78.*

Martin Wattenberg. 2006. Visual exploration of multivariate graphs. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, pages 811–819. *Cited on page 153.*

Martin Wattenberg and Danyel Fisher. 2004. Analyzing perceptual organization in information graphics. *Information Visualization*, 3:123–133, March. *Cited on page 39.*

Martin Wattenberg and Fernanda B. Viégas. 2008. The Word Tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 14(6):1221–1229, Nov./Dec. *Cited on pages 21, 31, 43, 62, 63, 66, 74, 78, 86, 144, and 238.*

Martin Wattenberg, Fernanda Viégas, Jesse Kriss, and Matt McKeon. 2008. Many Eyes. Website, May. Available from: `http://www.many-eyes.com`. *Cited on page 10.*

Martin Wattenberg, Marek Walczak, and Jonathan Feinberg. 2000–2004. Apartment. Exhibited at the Whitney Museum of Art, London ICA, Ars Electronica, and online. Available from: `http://www.bewitched.com/apartment.html`. *Cited on pages 59 and 66.*

Chris Weaver, David Fyfe, Anthony Robinson, Deryck W. Holdsworth, Donna J. Peuquet, and Alan M. MacEachren. 2007. Visual exploration and analysis of historic hotel visits. *Information Visualization*, 6:89–103. *Cited on pages 11, 64, 65, and 74.*

Frank Webster. 2006. *Theories of the Information Society*. Routledge, London, 3rd edition. *Cited on page 20.*

Ben Werschkul and The New York Times. 2007. The 2007 State of the Union Address: The words that were used. Available from: `http://www.nytimes.com/ref/washington/20070123_STATEOFUNION.html`. *Cited on pages 55 and 68.*

Marcos Weskamp. 2004. Newsmap. Website, August. Available from: `http://www.marumushi.com/apps/newsmap/newsmap.cfm`. *Cited on page 66.*

Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2007. Scented Widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization). Cited on pages 145 and 153.*

James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. 1995. Visualizing the non-visual: spatial analysis and interaction with information for text documents. In *Proc. of the IEEE Symp. on Information Visualization*, pages 51–58. IEEE Computer Society. *Cited on pages 11, 28, 74, 78, 79, 80, 81, and 144.*

Jeremy M. Wolfe. 2003. Moving towards solutions to some enduring controversies in visual search. *Trends in Cognitive Sciences*, 7(2):70–76. *Cited on pages 14 and 35.*

Jeremy M. Wolfe, Anne Treisman, and Todd S. Horowitz. 2003. What shall we do with the preattentive processing stage: Use it or lose it? In *Proc. of the Annual Meeting of the Visual Sciences Society. Cited on page 35.*

Nelson Wong, Sheelagh Carpendale, and Saul Greenberg. 2003. EdgeLens: An interactive method for managing edge congestion in graphs. In *Proc. of the IEEE Symp. on Information Visualization*, pages 51–58. *Cited on page 230.*

Peter C. Wright, Robert E. Fields, and Michael D. Harrison. 2000. Analyzing human-computer interaction as distributed cognition: The resources model. *Human-Computer Interaction*, 15(1):1–41. *Cited on page 26.*

Naomi Yamashita and Toru Ishida. 2006. Effects of machine translation on collaborative work. In *Proc. of Computer Supported Cooperative Work*, pages 515–523, November. *Cited on page 109.*

Jing Yang, Matthew O. Ward, and Elke A. Rundensteiner. 2002. InterRing: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proc. of the IEEE Symp. on Information Visualization*, pages 77–84. *Cited on page 126.*

Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti A. Hearst. 2001. Animated exploration of dynamic graphs with radial layout. In *Proc. of*

the IEEE Symp. on Information Visualization, pages 43–50. *Cited on pages 214 and 217.*

Ji Soo Yi, Youn ah Kang, John Stasko, and Juile Jacko. 2007. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, 13(6):1224–1231, Nov./Dec. *Cited on pages 14 and 26.*

Jiajie Zhang and Donald A. Norman. 1995. A representational analysis of numeration systems. *Cognition*, 57:271–295. *Cited on pages 22 and 23.*

Torre Dana Zuk. 2008. *Visualizing Uncertainty*. Ph.D. thesis, University of Calgary. *Cited on page 40.*

Torre Zuk and Sheelagh Carpendale. 2006. Theoretical analysis of uncertainty visualizations. In Robert F. Erbacher, Johnathan C. Roberts, Matti T. Gröhn, and Katy Börner, editors, *Proc. of SPIE-IST Electronic Imaging*, volume 6060, 606007. *Cited on pages 14, 38, 39, 101, and 104.*

Torre Zuk, Lothar Schlesier, Petra Neumann, Mark S. Hancock, and Sheelagh Carpendale. 2006. Heuristics for information visualization evaluation. In *Proc. of the Conference on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, pages 55–60. ACM. *Cited on pages 38 and 39.*

INDEX