SlopeSeeker: A Search Tool for Exploring a Dataset of Quantifiable Trends

Alexander Bendeck
abendeck3@gatech.edu
Tableau Research,
Georgia Institute of Technology
Atlanta, Georgia, USA

Dennis Bromley
dbromley@tableau.com
Tableau Research
Seattle, Washington, USA

Vidya Setlur
vsetlur@tableau.com
Tableau Research
Palo Alto, California, USA

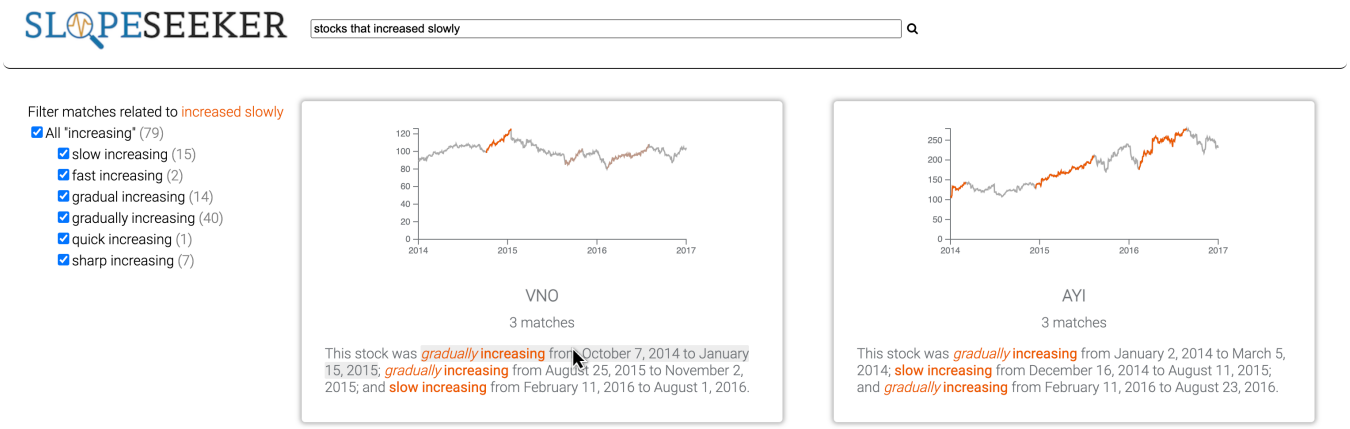

Figure 1: The SlopeSeeker interface. At the top, the search bar indicates a search for "stocks that increased slowly." Below that, the interface includes (from left to right) filter checkboxes showing labels related to "increased slowly" and two search result tiles, each showing a different stock price over time. The user is hovering over a text annotation in the left result tile. As a result, the tile's stock-description text is highlighted in gray, and the corresponding data trend on the stock price chart above is emphasized in red while other data trends on the chart appear faded.

ABSTRACT

Natural language and search interfaces intuitively facilitate data exploration and provide visualization responses to diverse analytical queries based on the underlying datasets. However, these interfaces often fail to interpret more complex analytical intents, such as discerning subtleties and quantifiable differences between terms like "bump" and "spike" in the context of COVID cases, for example. We address this gap by extending the capabilities of a data exploration search interface for interpreting semantic concepts in time series trends. We first create a comprehensive dataset of semantic concepts by mapping quantifiable univariate data trends such as *slope* and *angle* to crowdsourced, semantically meaningful trend labels. The dataset contains quantifiable properties that capture the slope-scalar effect of semantic modifiers like "sharply" and "gradually," as well as multi-line trends (e.g., "peak," "valley"). We demonstrate the utility of this dataset in SlopeSeeker, a tool that supports natural language querying of quantifiable trends, such as "*show me stocks that tanked in 2010*." The tool incorporates novel scoring and ranking techniques based on semantic relevance and visual prominence to present relevant trend chart responses containing these semantic trend concepts. In addition, SlopeSeeker provides a faceted search interface for users to navigate a semantic hierarchy of concepts from general trends (e.g., "increase") to more specific ones (e.g., "sharp increase"). A preliminary user evaluation of the tool demonstrates that the search interface supports greater expressivity of queries containing concepts that describe data trends. We identify potential future directions for leveraging our publicly available quantitative semantics dataset in other data domains and for novel visual analytics interfaces.

CCS CONCEPTS

• **Human-centered computing** → **Visualization systems and tools**; **Natural language interfaces**; • **Information systems** → *Specialized information retrieval.*

KEYWORDS

Semantics, search, trends, quantifiable metadata, visual analysis.

ACM Reference Format:
Alexander Bendeck, Dennis Bromley, and Vidya Setlur. 2024. SlopeSeeker: A Search Tool for Exploring a Dataset of Quantifiable Trends. In *29th*

IUI '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0508-3/24/03
https://doi.org/10.1145/3640543.3645208

International Conference on Intelligent User Interfaces (IUI '24), March 18–21, 2024, Greenville, SC, USA. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3640543.3645208

1 INTRODUCTION

Trends are patterns in data that indicate the general change in an attribute with time [13]. Searching for trends over time is a prevalent task in data analysis tools to identify anomalies or deviations from the normal or expected values in a dataset [6, 18, 34, 40, 54]. Trend analysis has significant relevance across application domains, ranging from discerning stock market trajectories and economic fluctuations to studying climate patterns, urban growth patterns, and monitoring disease epidemiology and health behavior [57].

During the height of the COVID-19 pandemic, for example, trends in the number of confirmed virus cases were constantly analyzed and compared between different geographic regions and time frames to understand the nature of the virus and the impact of different public health policies and mitigation measures. Enabling users to search for these types of trends using natural language (NL) would allow them great power and flexibility to express their intents. The difference between a "slow increase" and a "rapid increase" in COVID-19 cases could have huge implications for public policy; similarly, the difference between a stock price "slumping" and "tanking" is likely to invoke drastically different investment decisions. The expressive power in these scenarios comes from the precise, quantified semantics of these words used to describe the trends, and we argue that there is an opportunity for search tools to leverage these semantics to interpret expressive user analytical intents. Specifically, a set of quantifiable semantic labels can provide useful metadata to necessitate a structured approach to indexing, classification, and retrieval of trends in a search tool. Metadata that can encapsulate language that describes slopes and angles can further enhance the precision and recall of trends in search tools.

Search tools have recently evolved beyond document search, supporting intents for data exploration and providing results that include visualizations or widgets displaying data relevant to the user's query [50]. Similarly, natural language interfaces (NLIs) for visual data analysis [3, 7, 19, 24, 49, 51] now enable users to engage with and query data using NL. However, both these search tools and NLIs support only basic analytical intents, with limited support for interpreting temporal trends [9].

Contributions. In this work, we explore the potential of allowing users to search for temporal phenomena in a dataset by leveraging precise, quantified semantics of language, focusing on searching for trends in time series data. Specifically, our contributions are as follows:

- We collect a comprehensive dataset of semantic concepts describing trends and their quantifiable properties through crowdsourced data collection experiments. Going beyond prior work in this space [10], our dataset maps numeric slopes to semantic trend descriptor words and phrases; for example, we include slope labels (e.g., "falling") and slope labels with modifiers ("slowly falling"), along with multi-line trends that comprise a combination of "up," "down," and "flat" trend segments (e.g., "peak," "valley"). We release the quantified semantic trends dataset publicly[1]. Based on this dataset, we also introduce an approach for applying semantic trend descriptor labels to raw time series data.
- To demonstrate the applicability of our semantic trend labels dataset, we present the SlopeSeeker tool, which implements a novel analytical search experience supporting diverse trend search intents for these labeled trends. SlopeSeeker incorporates novel scoring and ranking techniques based on both the label relevance and visual prominence of trends. The tool also surfaces a semantic hierarchy of trend descriptor terms from our dataset, with which the user can interact to filter results.
- Using the SlopeSeeker tool as a design probe, we conduct a qualitative study with 12 participants to gain feedback on the trend querying features in the tool, the system design and implementation behavior, and how the labeled dataset aids in returning relevant trend search results. The study verifies that the trend-focused search paradigm effectively supports the distinct objectives of searching for pre-identified trend patterns. Finally, drawing from our observational data and participant responses, we identify potential directions for further development of the tool and the underlying semantic labeled dataset.
- We identify and discuss promising directions for future work in this space, such as employing LLMs for data augmentation and generating trend narratives, as well as exploring time normalization and predictive analytics for more nuanced interpretations of trend patterns.

2 RELATED WORK

Prior research relating to search systems in the context of visual data analysis falls under three main themes: (1) general search systems, (2) visual query systems, and (3) NLIs for visual analysis.

2.1 General Search Systems

Broadly, general search systems can be categorized into three types: those built upon structured query languages [16, 23, 27, 46], those that utilize keywords [14, 15, 26, 37, 61], and those that are based on natural language processing [15, 17, 21, 38, 39]. Our research extends the capabilities of general search to support intents that involve trends and their quantitative properties, i.e., slope and angle in line charts.

Early research in this area primarily aimed at enhancing conventional text search by integrating metadata using ontological methods to boost both recall and precision [8, 11, 25, 43]. More recent research has incorporated metadata related to attributes within curated data sources (e.g., synonyms and interrelated concepts) and metadata characterizing pre-existing content, such as visualization techniques, data attributes, and authorship [50]. In this work, we crowdsource data on the quantifiable semantics of trend descriptor words, specifically focused on capturing the interplay between language and slope, as well as multi-line trend shapes. This data is leveraged as metadata to boost precision and recall in the context of a search tool for retrieving relevant trend results.

[1]Link: https://osf.io/yzdvt/?view_only=d3723224f9234776a10882eee8b7568a

To optimize Q&A functionality in semantic search, various systems have been developed to precisely identify NL patterns. Common strategies combine statistical methodologies, such as syntactic parsing with semantic processes, to detect ontology-based concepts within user input. For example, QUERIX [32] integrates the Stanford CoreNLP parser with WordNet to discern prominent NL phrases in user queries [35]. Other Q&A frameworks employ linguistic analysis to identify pertinent entities and phrases [55, 58]. SlopeSeeker detects analytical trend intents in the search queries and finds trends matching the specified quantifiable properties such as "sharp decline" and "gradual rise" in univariate line charts.

2.2 Visual Querying Systems

Visual querying systems [36] are specifically designed to simplify the process of identifying desired visual patterns within datasets. The ZenVisage [52] visual analytics system was designed for this purpose. Hochheiser & Shneiderman [28] developed a visual query system for time series data, which relied on an interaction method of a "time box" by specifying a rectangular region spanning a range of both time and value. Time Lattice supports interactive analysis by customizing a data-cube structure for time-series data with an implicit temporal hierarchy [42]. Zhao et al. employed a KD-tree to speed up temporal queries to assist analysts in exploring the local pattern details of interest [62]. However, these systems do not enable users to employ NL for perusing and exploring the data, nor does it have the capability to understand quantified semantics in trends and patterns. Lee et al. subsequently further explored the space of visual search for data and identified the need for expressive querying and faceted exploration [36]. Our work focuses on NL input as a modality for users to express trend patterns in search, along with faceted browsing to drill up and down the hierarchy of semantic concepts describing these trends.

Continuing this theme of research, Siddiqui et al. introduced ShapeSearch [53] to facilitate the search for specific patterns through sketching, NL, and visual regular expressions. However, the tool does not support the interpretation of quantified semantics for trend descriptors. While basic descriptors like "up" or "down" are supported, the tool does not accommodate variations in slope and magnitude properties present in trend patterns. In addition, there is no integration of text with the charts to provide additional context to the user during their search task. Our work further explores the nuances of trend patterns and their properties using NL as the modality for expressing such queries. We also integrate text with the search results, along with faceted browsing, to provide additional information and expressivity for navigating the search results.

Bromley and Setlur [10] recently established an approach for labeling semantic visual features in line charts and proposed its use in supporting the search of shape descriptors for trends. We further improve upon this work by carefully designing our experiments and curating our dataset to leverage the precise semantics of trend descriptor words, trend descriptors with modifying adverbs, and multi-line shape trends. In particular, our experiments are aimed at understanding nuances between singleton slope labels (e.g., "falling"), slope labels with modifiers ("slowly falling"), and multi-line trends that comprise sequential combinations of line segments (e.g., "peak," "valley").

2.3 NLIs for Visual Data Analysis

NLIs for visual analysis are designed to facilitate analytical Q&A [3, 7, 24]. They generate charts based on inferred user intent and subsequently introduce ambiguity widgets, allowing users to modify predefined system selections. Both Eviza [49] and Analyza [19] operate upon this premise by integrating contextual inferencing capabilities. Other systems, such as Evizeon [31] and Orko [56], explore the support of pragmatics within an analytical conversation, leveraging an understanding of the conversational context in play. Flowsense enables NL-based interactions in a dataflow system [60]. However, the scope of these NLIs focuses on the general support of analytical inquiry and does not consider the interpretation of intents specific to trends and their semantic concepts.

The iGraph system [22] focuses on querying trends observed in line graphs; however, the linguistic model employed in the system provides limited support for querying trends and their semantic features pertaining to their quantifiable properties in their slope features. Hoque et al. presented a comprehensive overview of the existing landscape of chart question-answering systems [30]. Their survey identified opportunities for supporting more open-ended queries for visual representations and the use of language and semantics to provide more sophisticated models for Q&A support for data exploration. More recently, the Olio [50] hybrid search system combined semantic Q&A search with document-based exploratory search over data repositories. While the system supports basic analytical intents such as groupings, filters, geospatial, and temporal queries, there is limited support for querying specific semantics for quantifiable concepts such as "gradual," "sharp," or "plateau" patterns in trends. Our work further builds upon these search and NLI systems to support the exploration of trends with a comprehensive labeled semantic concept map of trends and their properties.

3 CREATION & UTILIZATION OF QUANTIFIED SEMANTIC LABEL DATASET FOR TRENDS

To support an expressive analytical experience for exploring relevant trends in a time series dataset, having a quantifiable understanding of the semantics of the trends can be useful. For example, while describing a stock price as "slumping" intuitively corresponds to a less severe decline than "crashing," quantifying the nuanced differences between these terms will enable visual data analysis tools to more easily leverage the words' expressive power. Bromley and Setlur [10] previously proposed a crowdsourced dataset of quantifiable visual features for supporting the search of trend shape descriptors.

However, our experiment design and subsequent analysis extend beyond their work and address shortcomings in several key areas:

- **Greater precision in quantifying slope semantics.** In our experiments, we ask participants to directly label isolated slopes with trend descriptor labels. In Bromley and Setlur's work, participants were instead asked to label line charts (rather than slopes), making it difficult to isolate the direct correspondence between labels and quantifiable slopes. For

instance, a participant may have labeled a particular chart with the word "soaring," but each chart contained multiple line segments, making it difficult to ascertain exactly which line segments the participant considered to correspond to a "soaring" slope.

- **Identification of nuances between trend descriptors.** We carefully account for and quantify the effect of modifying adjectives or adverbs (e.g., "fast") on trend descriptor verbs (e.g., "falling") when describing trends. By contrast, Bromley and Setlur treated trend descriptor verbs and modifying adjectives/adverbs as equivalent entities, ignoring the nuances of how these word types can interact to change the semantics of a quantified trend description.
- **Support for multi-line shape descriptors.** We take a thoughtful approach in dealing with multi-line segment shapes in the data (e.g., "peak" or "valley"). In Bromley and Setlur's work, these shape descriptors were treated as equivalent to slope descriptor verbs and associated with a single slope.
- **Consideration of semantic relationships between words.** Our analysis of collected trend descriptor word data includes a discussion of suggested semantic relationships (synonym, hyponym, and hypernym relationships) between trend descriptor words. Bromley and Setlur simply treated all words as a flat list with slopes assigned along a continuum.

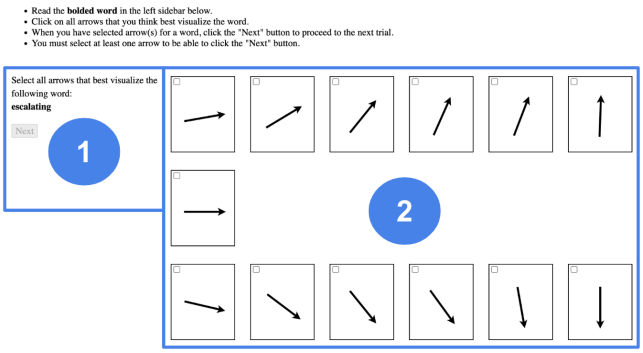

Figure 2: The interface for the data collection web tool used in Experiment 1. (1) The participant is prompted with a word and asked to select all arrows that best visualize the word. Once complete, the participant can click the "Next" button to proceed to the next word. (2) The participant is shown 13 arrows corresponding to an array of angles between -90° and 90°. Clicking anywhere inside an arrow's box applies the current word as a label to the clicked arrow. Note that the interface is similar in Experiment 2, except that in (1), participants are first shown an individual anchor word and then four compound labels, which can be assigned to arrows in turn.

3.1 Quantified Semantic Label Dataset Collection & Analysis

To achieve the advances mentioned above, we designed and conducted three crowdsourced experiments to collect a dataset of quantified semantics for trend descriptor words. We recruited all participants through an internal mailing list at an analytics software company.

We first collected a set of 41 words used to describe trends that can be mapped to a single quantified slope. We started with the initial list of 21 verbs from Bromley and Setlur's work, then removed 5 words (namely "accelerating," "intensifying," "decelerating," "subsiding," and "bouncing") that we deemed could not be mapped to single slope values. We subsequently added six nouns and adjectives from Bromley and Setlur's work that correspond to flat slopes (namely "flatline," "plateau," "stagnant," "constant," "stable," and "even"). Finally, we augmented this list with 19 additional words sourced from GPT-4 [45] and WordNet [41] by querying for synonyms of the words already on the list. The synonyms suggested by GPT-4 were manually inspected by the authors to filter out hallucination responses. Synonyms were only included in the final list if deemed appropriate by all authors.

In the final set, 17 of these words correspond to negative slopes in time series data (e.g., "falling," "dropping"), 14 correspond to positive slopes ("growing," "rising"), and 10 correspond to relatively flat slopes (see examples above). These words collectively formed the word corpus used for Experiments 1 and 2. For Experiment 3, we included four words from Bromley and Setlur's work that described multi-segment shapes rather than individual slopes (e.g., "peak," "valley"). We also included 14 additional multi-segment words from GPT-4 with an input prompt, "*what are the most common multi-segment words similar to 'peak' and 'valley'?*," for a total of 18 such words. As before, these words from GPT-4 were checked for appropriateness by the authors. The data collected from all experiments is included as supplemental material.

3.1.1 Experiment 1: Quantifying Precise Slope Semantics for Individual Trend Descriptors.

Method. In Experiment 1, our goal was to collect slope information for each of our 41 single-slope trend descriptor words. The tool's interface (Figure 2) presented a set of 13 arrows whose slopes ranged from 90° to -90° (straight up to straight down, respectively) in increments of 15°. Slopes were jittered by ±7° to provide diversity in the labeled angles. The 7° jitter (just under half of the angle increment between adjacent arrows) was chosen so that the order of the arrows being arranged with increasing magnitude from left to right across the screen would always be preserved. We also made sure that jittered slopes would not exceed the 90° or -90° boundaries since an arrow with such a slope would point backward and thus not make intuitive sense for correspondence with time series data.

The experiment was presented as a sequence of trials, each with one trend descriptor word, rather than as a single page with all 41 terms present at once. Our reasoning was that we wanted to reduce cognitive burden and avoid visual clutter. For every trial, the participants were shown one of the trend descriptor words (e.g., "declining") and were asked to select all of the arrows whose slopes they felt best described that word. The trials were randomized

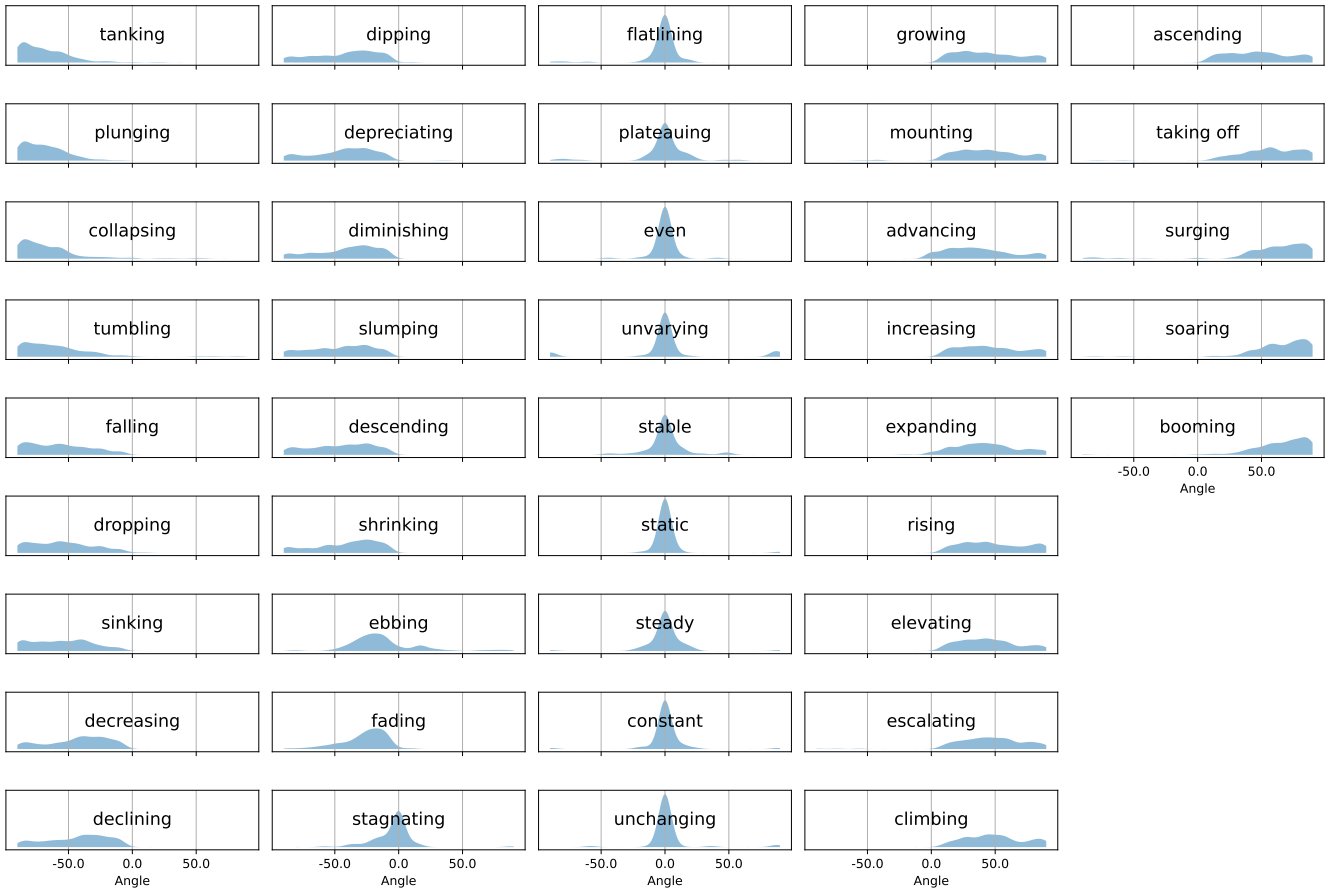

Figure 3: Experiment 1: One-dimensional KDEs indicating probability density for each label over the range of -90° to 90°. Peak probability density was used to sort the labels from the most negative angle (steepest down) to the most positive angle (steepest up) from the top left ("tanking") to the bottom right ("booming"), respectively. Note that the distributions are not normal.

for each participant to mitigate the effects of word presentation order and ensure close to even labeling coverage across words, even if participants did not complete all trials. When a participant clicked an arrow to apply a word label, we recorded the arrow identifier, the arrow slope, the word label, a timestamp of when the annotation occurred, and a unique anonymous participant identifier in a PostgreSQL database [48].

Analysis. 80 participants participated in Experiment 1. Overall, 5,346 angle labels were collected for an average of 67 labels per participant. Across the 13 arrow positions (see Figure 2), there was an average of 421 (min=301, max=531) labels and 21 (min=16, max=29) unique labels per arrow position. The data was analyzed with the goal of estimating a slope distribution for every label. We employed Kernel Density Estimation (KDE) [47] as a technique for estimating the slope distribution for each label, as KDE is a common tool for estimating the probability density function of a random variable without making assumptions about the nature of the distribution. The Gaussian kernel is a common choice for smoothing KDE data points as the shape is symmetric, it has well-understood mathematical properties, and, notably, the "bandwidth" KDE parameter can be interpreted as the Gaussian standard deviation. We selected a bandwidth parameter of 5° to balance between under-fitting and over-fitting the per-label angle data – a reasonable scale for the range being analyzed (-90° to 90°). Figure 3 shows the slope distribution of each label. It is worth noting that final slope labeling is not particularly sensitive to choices of bandwidth value. Final labeling is performed by taking the KDE data point with the highest probability density at a given angle (i.e., slope). Since we use the same bandwidth parameter for all data points, changing the bandwidth would cause all Gaussian distributions to move up and down together, changing the absolute density values but not changing the stack-ranked order. Thus, reasonable changes in bandwidth should not affect the "winning" label at any point in KDE space.

3.1.2 Experiment 2: Identifying Nuances Between Trend Descriptors with Modifiers.

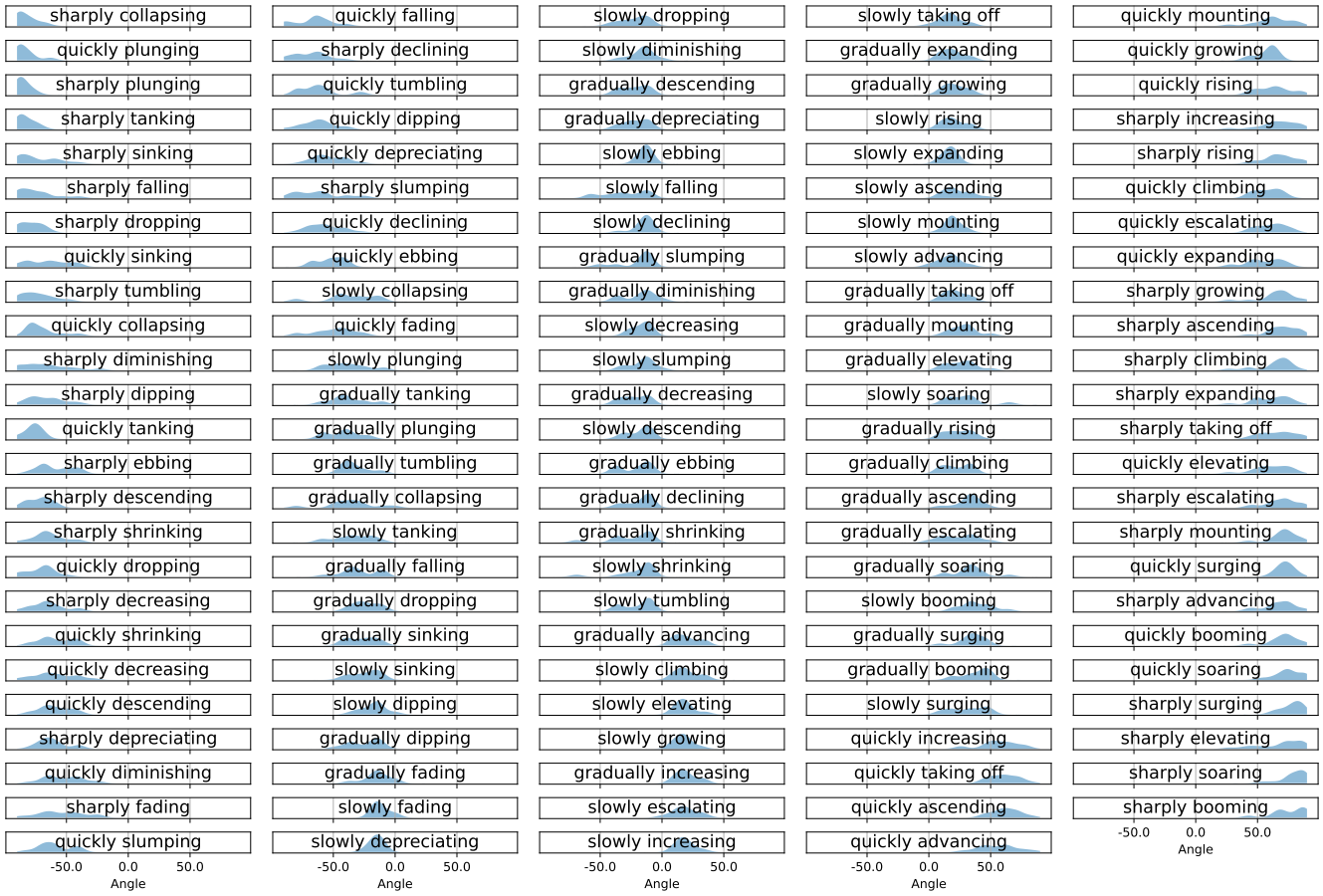

Figure 4: Experiment 2: One-dimensional KDEs indicating probability density for each label over the range of -90° to 90°. Peak probability density was used to sort the labels from the most negative angle to the most positive angle from the top left ("sharply collapsing") to the bottom right ("sharply booming"), respectively. Note that the distributions are not normal.

Method. In Experiment 2, our goal was to assess the impact of modifier adverbs on the quantified semantics of two-word trend descriptor phrases, e.g., "*slowly* falling." Of the original set of 41 words from Experiment 1, we excluded 10 words corresponding to flat slopes since we would not expect to use modifiers to change a label like "constant" or "even" to subsequently refer to a different, non-zero slope. We thus only considered the remaining 31 words that corresponded to positive or negative slopes. To create two-word phrases, we selected two adverbs that would make slopes more extreme ("quickly," "sharply") and two that would make slopes less extreme ("gradually," "slowly") compared to the anchor label.

For each trial, participants were first shown a verb and asked to select a single arrow whose slope they felt best matched the word. This single slope was then used as the label "anchor" so that participants had a fixed point of reference for the slope labels with modifiers. The participant was then shown four compound labels with the anchor word and the modifiers (e.g., "quickly falling," "sharply falling") and asked to select all arrows that were described by each label. Participants could also choose to discard a compound label if they did not find it to be meaningful (e.g., "slowly tanking"). Participants were restricted from assigning a compound label to the same slope arrow as the anchor word itself, since we expect the modifying adverbs to always have some effect on the slope of the verbs they modify. In all other cases, a single slope could be assigned multiple compound labels (e.g., a single slope could be both "quickly tanking" and "sharply tanking"). Similar to Experiment 1, the trials were randomized such that anchor words were presented to each participant in a random order. The compound labels with modifiers were listed in a randomized order for each participant (to mitigate bias) but not for each trial to avoid disorienting participants. Whenever a participant clicked an arrow to apply a word label, we recorded the same data as in Experiment 1 with the addition of the current modifier word (or an empty string if the user was setting an anchor arrow).

Analysis. 37 participants participated in Experiment 2. A total of 2,005 labels (excluding anchor words) were collected. Five participants only picked anchor words; their data was implicitly excluded

as there were no modifiers to analyze. The remaining 32 participants had an average of 62 (min=1, max=196) labels per participant. Of the total labels, 144 were "trashed" label modifications that participants did not deem reasonable (e.g., "*slowly* plunging"). Of the 31 anchor labels, 28 labels were trashed for at least one modification ("rising," "sinking," and "climbing" did not have any trashed modifiers). However, all of these trashed labels were assigned angles by other participants, so there was no general consensus on what label modifications were unreasonable. The remaining 1,861 non-trashed label modifications were spread across all 13 arrow positions (see Figure 2), resulting in an average of 157 (min=13, max=204) total label modifications and 33 (min=11, max=49) unique label modifications per arrow position. The following rules were used to clean the crowdsourced label data from Experiment 2. In total, 93.4% of label/modifier pairs passed these tests and were used for subsequent analysis:

(1) Remove label/modifier pairs where the $\frac{\text{modifier}}{\text{anchor}}$ ratios for "slowly" and "gradually" were > 1.0 or "quickly" and "sharply" were < 1.0; these values were deemed to not be consistent with the common semantic meaning of those words (e.g., the modifier "*slowly*" should not make the slope of "falling" more extreme).
(2) Remove label/modifier pairs where the anchor angle against which the modifiers were compared was 0°; calculating this scalar $\frac{\text{modifier}}{\text{anchor}}$ ratio led to a divide-by-zero error.

Slope analysis of compound labels (e.g., "*slowly* falling") was identical to the analysis of singleton labels in Experiment 1. Figure 4 shows the KDE distributions of Experiment 2's compound labels.

The data from this experiment also enabled the computation of the overall scaling effect each modifier adverb had on each label's associated angle. For example, consider a data point from a single participant indicating that the "anchor" angle associated with the label "dropping" is approximately -48°. A subsequent data point from the same participant indicates that the angle associated with the compound label "*sharply* dropping" is -88°. In this case, we can calculate the scalar effect of the modifier "sharply" to be -88/-48 = 1.8. The slope difference between "dropping" and "*sharply* dropping" can thus be quantified: "sharply dropping" is 1.8 times steeper than simply "dropping" for this participant. We collected these ratios for all angle/modifier data points across all participants and, as before, used KDE analysis to estimate a scalar distribution for every modifier. We again used a Gaussian kernel, and for this analysis, we used a bandwidth parameter of 0.1 to balance between under-fitting and over-fitting the data. As shown in Figure 5, in general, "slowly" reduces slope steepness by a factor of 0.4, "gradually" reduces slope steepness by a factor of 0.6, and "quickly" and "sharply" *increase* slope steepness by factors of 1.3 and 1.5, respectively.

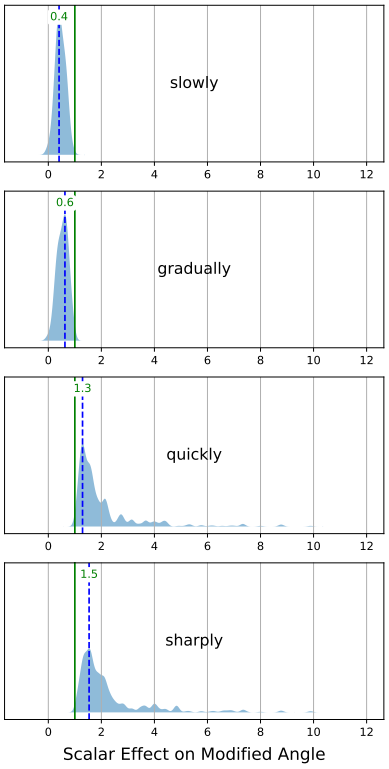

Figure 5: One-dimensional KDEs indicting the scalar range over which different label modifiers scaled the base angle of the label. The solid green line indicates the 1.0 line, and the labeled dotted blue line indicates the scalar value at the peak probability density. Notice that "slowly" and "gradually" have scalar values between 0 and 1 (0.4 and 0.6, respectively), i.e., they reduce the steepness of a label's angle, while "quickly" and "sharply" have scalar values greater than 1 (1.3 and 1.5 respectively), i.e., they increase the steepness of a label's angle.

3.1.3 *Experiment 3: Supporting Multi-Segment Shapes.*

Method. In Experiment 3, we aimed to gather labels for different shapes found in time series data. Given that such shapes, in general, can be arbitrarily complex, we focused on performing a relatively thorough sampling of the space of simple shapes consisting of two line segments (e.g., peaks, valleys, plateaus, up-ramps, down-ramps, etc. – see top inset in Figure 6). We define a *shape* as a pair of connected line segments with varying degrees of (1) inclination angle between the two lines and (2) overall 360° rotation or orientation. Using a similar web interface as before (see Figure 6), participants were asked to label shapes by dragging words from a word list onto the shapes they felt best matched that word. Shape angles spanned the range of 0-180°, and shape rotation spanned the range of 0-360°. Some angle/rotation combinations resulted in non-monotonic shapes; these were removed from the interface (see Figure 6). Participant data was again collected in a PostgreSQL database. Word lists and shapes were randomly arranged to avoid positional bias.

Analysis. We collected 347 labels from 24 participants for an average of 14 shape labels per participant. The average shape was assigned six different labels (min=3, max=9), e.g., one shape was assigned the labels "valley," "trough," "spike," and "crash." Conversely, the average label was used to describe eight different shapes (min=4, max=15), e.g., "uptick" was assigned to 12 different shapes. Similar to Experiments 1 and 2, KDEs were used to describe the label/shape

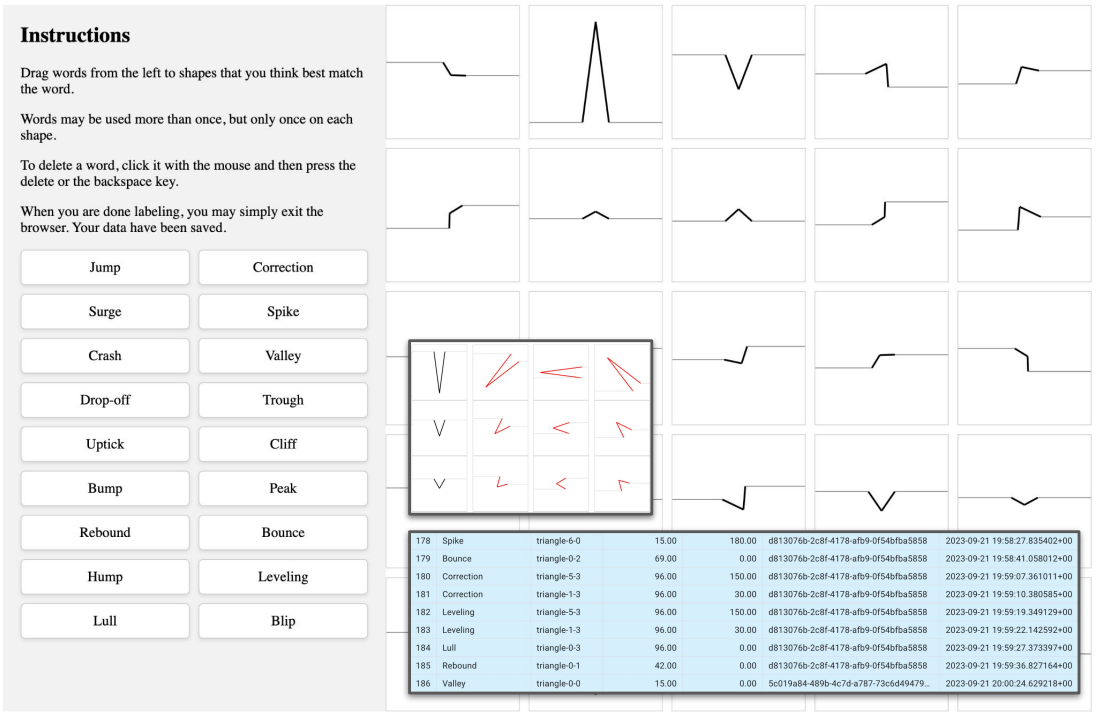

Figure 6: Screenshot of the user interface for Experiment 3. Participants dragged descriptive labels from the left onto shapes on the right. Top Inset: Shapes were generated by transforming two-segment angles: angles became more obtuse from top to bottom and were rotated from left to right. Non-monotonic shapes (shown in red) were removed and not shown to the user. Bottom Inset: User labels were recorded in a PostgreSQL database; a snapshot of illustrative data rows is shown.

distributions. However, because shapes were parameterized by both angle and rotation, Experiment 3 used two-dimensional KDEs instead of one-dimensional KDEs, resulting in 2D density plots instead of 1D density plots like those from Experiments 1 and 2. We employed Gaussian KDE kernels with a bandwidth of 15°, an empirically derived balance between under-fitting and over-fitting the data that was a reasonable scale for the values being analyzed (0-360° for rotation and 0-180° for angle). Also, as discussed above, the final label selection is tolerant to a reasonable range of bandwidth values, so extreme precision is not a significant concern.

Since the shape rotation was periodic, the 2D shape KDEs were made periodic by wrapping them at the 0°/360° boundary. For example, say we wanted to know the data density at a rotation of 3°. To make sure we account for probability density from a point at 355°, we calculated the 3° mark's *virtual* point to be 360 + 3 = 363°, which is influenced by data at the 355° mark after the boundary wrapping. We then summed the data for the 3° mark and the 363° mark to calculate the final reported value. To calculate periodicity, the 0°/360° boundary overlapped by ±3*bandwidth = ±45°. The ±3*bandwidth overlap was chosen because three standard deviations (for Gaussian KDE, bandwidth = standard deviation) include 99.7% of the Gaussian distribution. Figure 7 shows the 2D KDE plots for Experiment 3.

Quantifying Semantic Relationships. We were also interested in exploring how the quantitative labeled data from the experiments could inform the creation of a semantic ontology akin to Wordnet, where semantic concepts are linked by various semantic relations such as synonyms and hypernym/hyponym (superordinate/subordinate) relations [41]. Such a structure could be useful for supporting faceted search behavior to drill down or up the semantic hierarchy.

Figure 8 shows a scatterplot of the median, mode, and IQR for the angle distribution of each Experiment 1 label. Wider IQRs are placed higher up, highlighting that the angle ranges of some labels could subsume the angle ranges of other labels, suggesting a hypernym-hyponym relationship (e.g., "rising" subsumes "climbing" and thus could act as a hypernym).

While the quantitative semantic data shown in Figure 8 suggest synonym or hypo/hypernym relationships that might be derived (*plateauing* >= *stable* >= *steady* >= ...), many of the relationships appear to be more nuanced and suggestive of either partial or multi-category hyponymy and hypernymy. For example, "ebbing" could perhaps be considered a partial hypernym of "fading;" there is a partial subsumption relationship, but at the extrema, "ebbing" suggests a more positive angle while "fading" suggests a more negative angle. Similarly, "tumbling" could be seen as a hyponym of *both* "falling" and "diminishing." While these statistical methods are

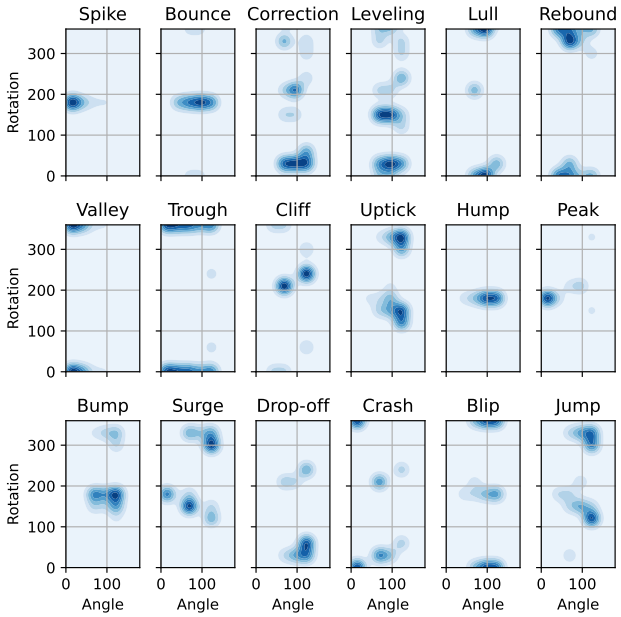

Figure 7: A grid showing a two-dimensional KDE plot for every shape label. Angles range from 0-180° along the x-axis and 0-360° along the y-axis. The y-axis is periodic; thus, the probability density is continuous across the 0°/360° border.

quite basic, they suggest that future quantitative semantic analysis could inform the automatic creation of semantic networks and ontologies.

Across the three experiments, we collected 7,554 crowdsourced labels for slopes (5,346 labels), adverb-modified slopes (1,861 labels), and shapes (347 labels), and the data is publicly available in the supplementary material.

3.2 Labeling Events in Time-Series Data

Once the KDE distributions for both slope labels and shape labels were established, we aimed to use these distributions to label new time series data represented as raw, univariate input signals.

At a high level, the algorithm for label assignment is as follows: decompose the input signal into linear segments, calculate angles and rotations over those segments, use those angles and rotations to index into the KDEs from the three experiments, and discover appropriate labels. Before we could execute these steps, however, we needed to account for a discrepancy between the aspect ratio of the slope arrows labeled in our data collection experiments and the stock data charts that we plan to use to display stock data in SlopeSeeker.

Say that in a chart with a 1:1 aspect ratio, a participant labeled a 45° angle (slope of 1.0/1.0 = 1.0) as "ascending." If we stretched that same chart to be 100x as wide as it was tall (aspect ratio = 1:100), the participant would probably label that slope as "flat." Conversely if we compressed the chart to be 100x as tall as it was wide (aspect ratio = 100:1), the participant would likely label that slope as "soaring." Thus, it seems reasonable to account for aspect ratio when labeling the perceived "steepness" of a line. In this example, we are analyzing a slope where the stock price increased 100% in 100% of the time – that is, it spanned an equal distance along both the x and y axes. In a chart with a 1:1 aspect ratio, this slope would be 1.0/1.0 = 1.0 = 45°, and we would label it "ascending" per the participant's input. However, when we present the chart *visually* to the user, we present it with an aspect ratio of approximately 3:1, *stretching* the x-axis such that the same $[0.0, 1.0]$ span looks three times longer on the x-axis than the y-axis. This leads to a *perceived* slope of 1/3 = 0.333 = 18.4°. Thus the user actually sees a line with a slope of 18.4°, not 45°, and the "ascending" label looks incorrect because an 18.4° slope, according to our crowdsourced data in Figure 3, is closer to "growing" or "mounting" than to "ascending." The core issue is that the participant labeled a *perceived* 45° angle as "ascending" and then we showed them a *perceived* 18.4° angle and labeled it (seemingly incorrectly) as "ascending" because we did not compensate for the aspect ratio of the visual presentation. While we realize that there is no "correct" data transform as such for this scenario, it seems reasonable that the labeled angle from the input tool should look like the labeled angle in the visual output. Without accounting for the aspect ratio of how the data is presented, the angle and rotation calculations would thus not be true to our collected crowdsourced data. To resolve this issue, we made two observations:

(1) Participants labeled angles in a user interface with an aspect ratio of 1:1, meaning that the visual space was "square."
(2) The quality of "steepness" is, to a large degree, perceptual and anchored to both the time range we are analyzing and the shape (aspect ratio) of the *chart* and how lines are *drawn* on that chart. Note that this has nothing to do with display size or display device configuration or resolution; this aspect ratio correction is necessary because these labels are *perceptual* labels, not absolute data labels, and as such, we need to correct for the perceived change in angle when a chart is compressed or expanded to an aspect ratio other than 1:1.

Our goal was to design an algorithm that would provide perceptually reasonable results; in particular, slope labels in the output visualizations should visually correspond to the slope labels in the data collection experiments. To this end, we transform the input signal in two ways before we perform the analysis. First, we normalize both the temporal measure (x-axis data) and the stock-price measure (y-axis data) to span the range $[0.0, 1.0]$, placing both measures on the same scale.

We then scale both measures by the aspect ratio of the expected visual presentation, in this case, 3:1. As a result, the y-axis spans the range $[0.0, 1.0]$, and the x-axis spans the range $[0.0, 3.0]$. This causes the $[0.0, 1.0]$ span of the x axis to be the *same* perceived distance as the $[0.0, 1.0]$ span of the y axis. Thus, a line that spans equal distances along the $x-$ and y-axes (say, 0.5 along both) will have a perceived slope of 0.5/0.5 = 1.0 = 45° and will be labeled "ascending" as expected.

Following this axis normalization, we proceeded to identify and label temporal stretches of raw input data. The first step was to decompose the input signal into consecutive linear segments using the Ramer-Douglas-Peucker signal linearization algorithm [20] as implemented in the *rdp* python package (https://pypi.org/project/rdp/). Epsilon values for the Douglas *et al.* algorithm, which control

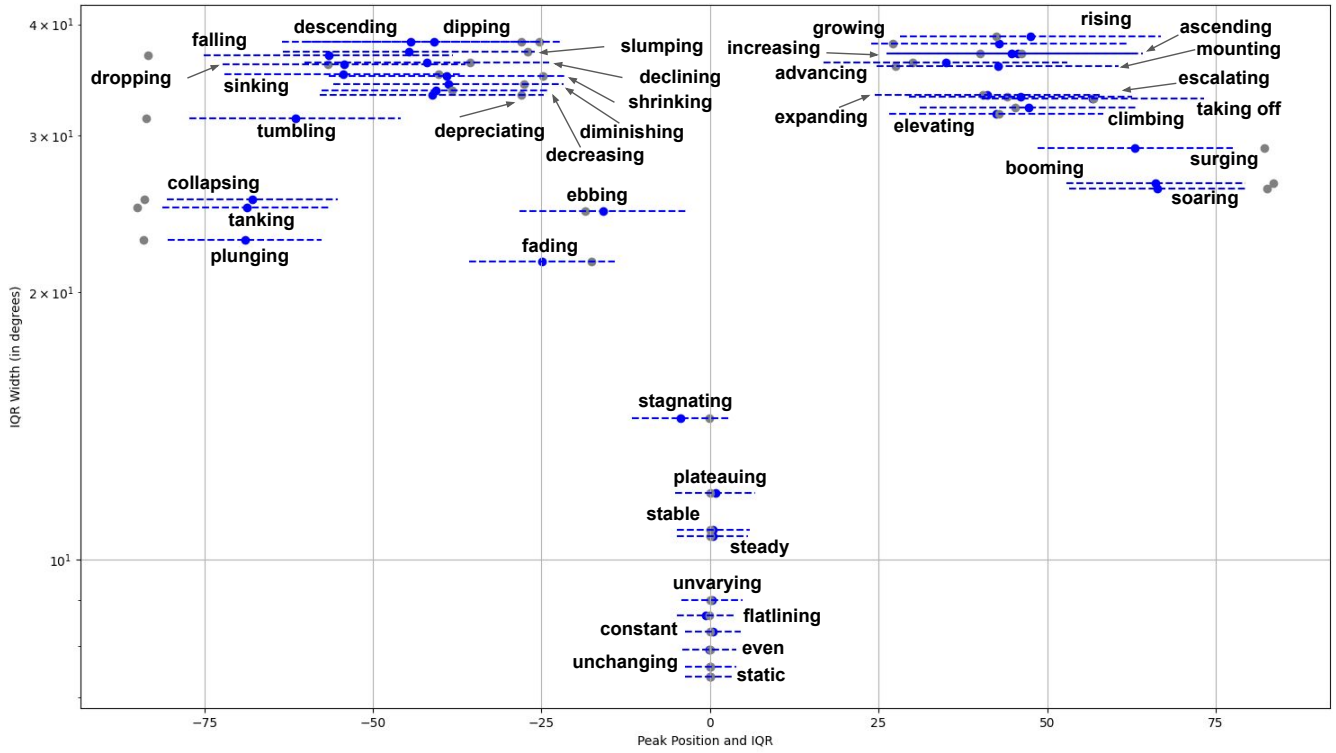

Figure 8: A scatter plot suggesting implicit semantic hierarchies. The x-axis shows the angle range to which the label has been assigned by experiment participants, and the y-axis indicates the width of the inter-quantile range (IQR) of the label angle distributions; labels with a broader definition (applicable to more angle ranges) are at the top, labels with narrower definitions are at the bottom. The blue lines indicate the IQR for each label, the blue dots indicate the center of the IQR, and the gray dots indicate the location of the peak value (mode). Note that the label distributions are not normal (see Figure 3), so the mode may lie outside of the IQR.

allowable linearization error and thus the length and scale of the linear stretches, were empirically chosen as 0.03, 0.1, and 0.2 to provide three different linearization resolutions relevant to the stock data under analysis (these values could be adjusted for specific datasets and analyses).

We then determined labels for these linear segments. For single-segment temporal stretches, we first calculated the slope of the segment. We used that slope to index into all the single-label (Experiment 1) or compound-label (Experiment 2) 1D KDEs. The label whose KDE returned the highest probability density was chosen as the label for that segment. Since the KDE models are a continuous surface (1D for Experiments 1 and 2, and 2D for Experiment 3), any data point (i.e., line segment) – even one that is far away from all labels – will always return a non-zero probability density score, resulting in *some* (possibly inappropriate) label. To resolve this issue, we took the set of all segment labels (one label per segment), sorted them by their probability density, and only used the top-scoring 75% of labels for the SlopeSeeker database. Examples of segment labeling with single-label and compound-label are shown in Figure 9 and Figure 10, respectively. For two-segment (shape) temporal stretches, the process was very similar. We first calculated the angle and rotation of each shape. We then used the angle and rotation to index into all the 2D label (Experiment 3) KDEs and kept the top 75% of labels. Examples of shape labeling are shown in Figure 11.

To support the querying of superlative features within a trend (e.g., "maximum," "minimum," "highest point"), we additionally identify the highest and lowest values over the length of the time series data. An event consisting of 15 days before and after the maximum or minimum is subsequently incorporated into the event label to result in month-long labeled events that are visually easy to locate (see Figure 12).

3.3 Visual Saliency Scoring

Prior research has shown that line chart annotations that emphasize the most visually prominent features of the chart are more effective at helping readers glean meaningful takeaways [33]. To operationalize this concept during search, we establish a way to quantify the visual saliency of each labeled trend event. Otherwise, it would be difficult to identify the most relevant results for a given search query when several matching results could have the same labels based on slope. Consider the simple case in Figure 13; while both events match a user query of "gradually increasing" based on slope, the event that occurred during 2016 intuitively appears more prominent and impactful than the event in 2015.

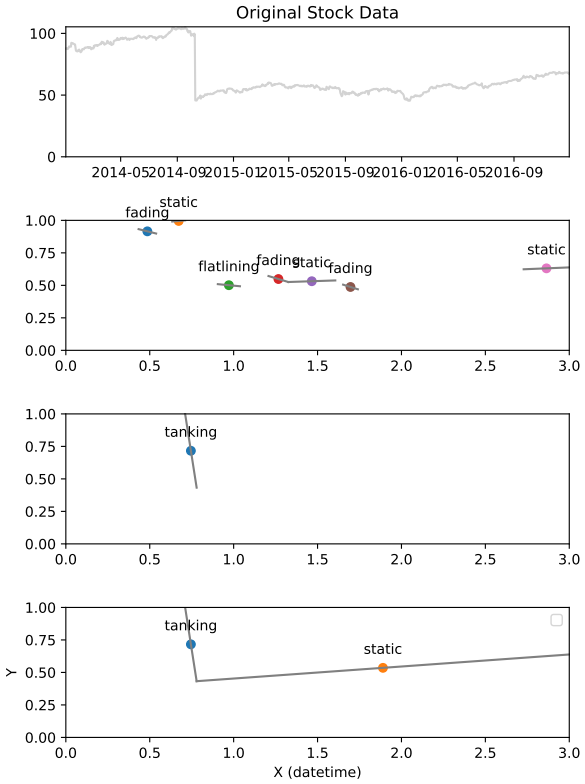

Figure 9: Experiment 1: Angle label assignment. The three sub-charts correspond to the three levels of linearization. The x axis indicates time, and the y axis indicates stock price. The original stock data (top) uses the original date and stock value for the x and y axes, respectively. The remaining three charts use normalized x and y values, scaled by the chart's 3:1 aspect ratio, resulting in an x range of $[0.0, 3.0]$ and a y range of $[0.0, 1.0]$. For clarity, only the top 25% of labels are shown.

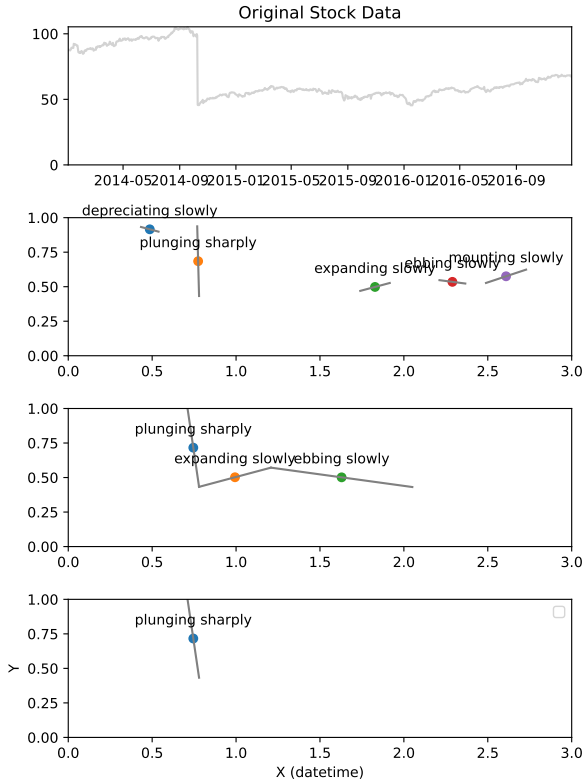

Figure 10: Experiment 2: Angle label assignment. The three subcharts correspond to the three levels of linearization. The x axis indicates time, and the y axis indicates stock price. The original stock data (top) uses the original date and stock value for the x and y axes, respectively. The remaining three charts use normalized x and y values, scaled by the chart's 3:1 aspect ratio, resulting in an x range of $[0.0, 3.0]$ and a y range of $[0.0, 1.0]$. For clarity, only the top 25% of labels are shown.

The intuition behind our visual saliency scoring approach is to view each trend event as a vector that covers some of the encompassing chart's visual space in both the x direction (i.e., the temporal range of the trend) and in the y direction (i.e., the value range of the trend).

Algorithm 1 Visual saliency computation algorithm

for each trend result (single-segment slopes): **do**

Compute the x vector component as the ratio of the entire time range taken up by the trend.

Compute the y vector component as the ratio of the entire data value range taken up by the trend.

Take these two vector components and use the Pythagorean theorem to compute the L2 norm.

end for

Formally, we use the following equation (or see Algorithm 1 for a procedural representation):

$$\sqrt{\left(\frac{x_{event_{end}} - x_{event_{start}}}{x_{chart_{max}} - x_{chart_{min}}}\right)^2 + \left(\frac{y_{event_{end}} - y_{event_{start}}}{y_{chart_{max}} - y_{chart_{min}}}\right)^2}$$

where $x_{event_{start}}$ and $x_{event_{end}}$ are the data values on the x axis at the start and end of the event (and likewise for $y_{event_{start}}$ and $y_{event_{end}}$). Using similar notation, $x_{chart_{max}}$ and $x_{chart_{min}}$ are the maximum and minimum data values on the x axis over the entire time period of the chart (and likewise for $y_{chart_{max}}$ and $y_{chart_{min}}$).

Intuitively, trends described by words like "tanking" will mostly be short in x, in which case the most visually salient results will have the largest change in y. On the other hand, we anticipate that trends described by words like "flatline" will have little change in y, and so their visual salience will mostly depend on the duration in x. However, an intuitive correspondence between trend labels and their visual span in x or y does not need to hold for our scoring to

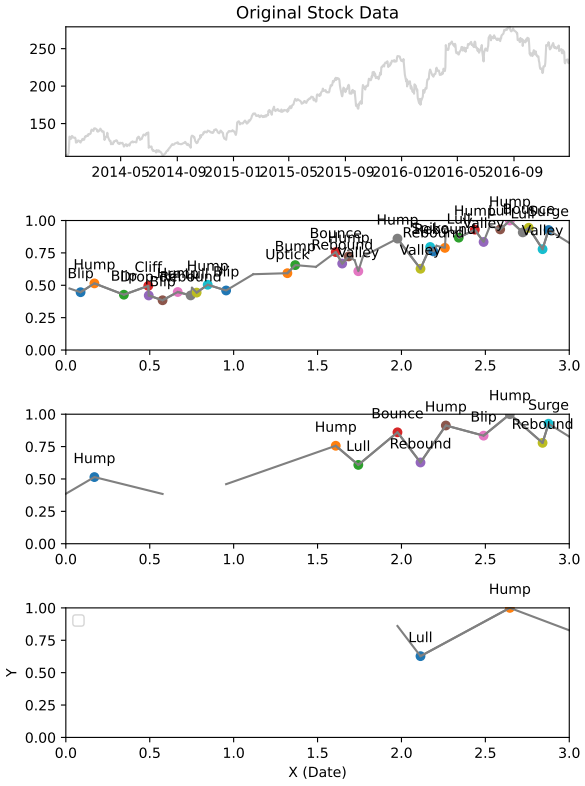

Figure 11: Experiment 3: Shape label assignment. The three subcharts correspond to the three levels of linearization. The x axis indicates time, and the y axis indicates stock price. The original stock data (top) uses the original date and stock value for the x and y axes, respectively. The remaining three charts use normalized x and y values, scaled by the chart's 3:1 aspect ratio, resulting in an x range of $[0.0, 3.0]$ and a y range of $[0.0, 1.0]$. For clarity, only the top 25% of labels are shown.

provide a useful quantification of visual saliency. Consider again the two events in Figure 13, both labeled as "gradually increasing". The event during 2016 is longer temporally (i.e., $x_{event_{end}} - x_{event_{start}}$ is greater) and spans a larger value range (i.e., $y_{event_{end}} - y_{event_{start}}$ is greater), therefore giving it a higher saliency score.

For multi-segment shapes, we compute the y vector component using the max and min values of y over the duration of the shape event rather than the start and end values. More precisely, we use $y_{event_{max}} - y_{event_{min}}$ in the equation above rather than $y_{event_{end}} - y_{event_{start}}$.

The final labeled stock data loaded into the SlopeSeeker tool contains 8,353 data points (labeled events) for 100 different stocks over a three-year period (2014 – 2016).

4 SLOPESEEKER TOOL

We developed SlopeSeeker as a search tool to operationalize our dataset of quantified semantic trend labels. In this section, we first

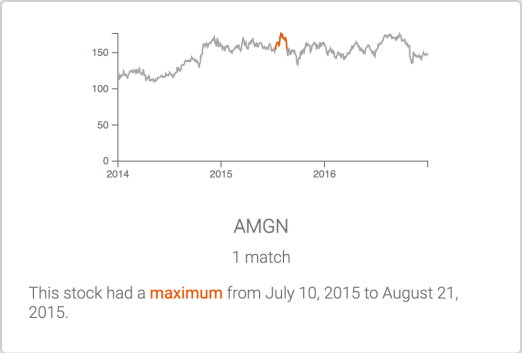

Figure 12: The highlighted event shows the global maximum for this stock over the entire time series.

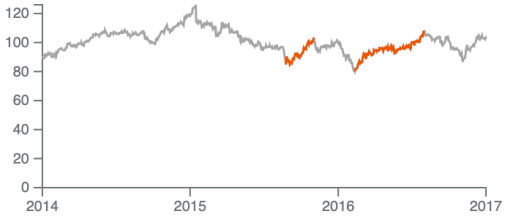

Figure 13: The two highlighted events have identical labels based on their slopes, but the event during the year 2015 is less visually salient than the event during 2016.

describe the tool's architecture and interface. We then detail the search framework underlying SlopeSeeker's functionality and how results are scored. We also outline the different types of trend queries supported by the tool – single trend events (e.g., "sharp increase" or "peak") and arbitrary event sequences (e.g., "up, down, flat").

4.1 Architecture Overview

SlopeSeeker is implemented as a web-based application using Python and a Flask backend [2] connected to a React.js frontend [4]. For data storage and retrieval, we employ Elasticsearch [1], a robust distributed search platform built on the open-source Apache Lucene. The platform offers scalability of data and real-time indexing for fast querying. We employ a RESTful API for easy integration with SlopeSeeker. Figure 14 illustrates the tool's architecture, with the following main components: a semantic parser (Section 4.3), a search index, and an interface manager that facilitates communication between these back-end components and the front-end interface to implement our end-to-end search framework (Section 4.4).

4.2 Interface

SlopeSeeker's interface (Figure 15) is designed to provide an experience similar to that of a common web search engine. The user is presented with a search box (Fig. 15.1), enabling them to enter

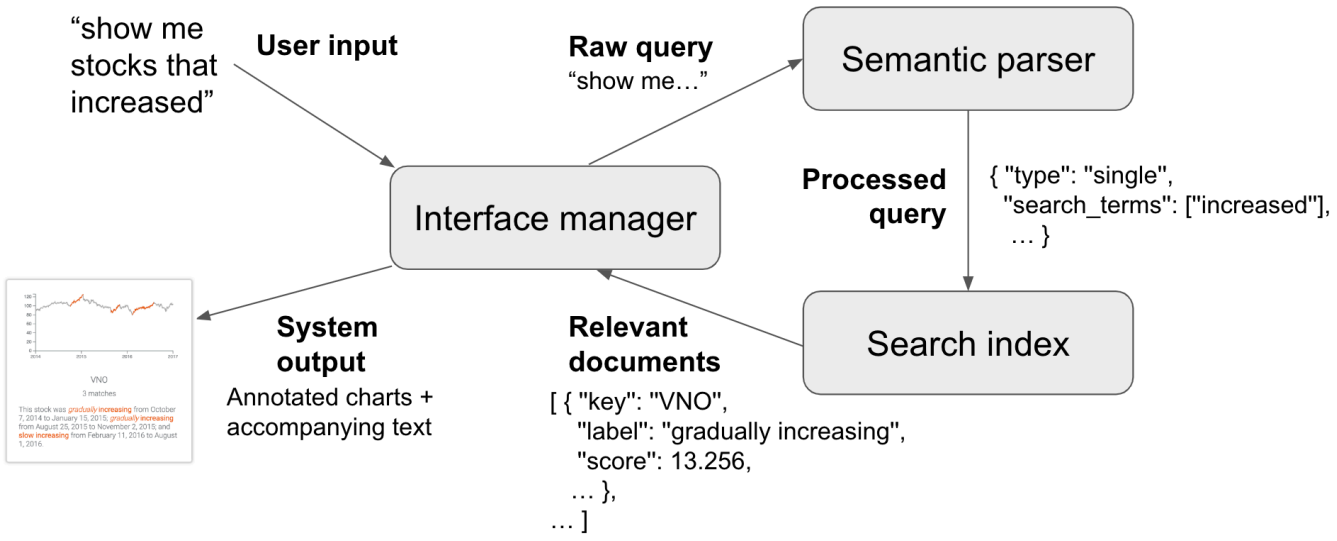

Figure 14: SlopeSeeker architecture overview.

a query to search for a trend of interest. After a query has been typed and the Enter key is pressed (or the magnifying glass search button is clicked), results appear as tiles below the search bar (Fig. 15.4). Each tile corresponds to one stock and shows a line chart of the stock price over time, the stock ticker, the number of matches for the input query for that stock, and text snippets (composed into a single sentence) describing up to the three highest-scoring matches for the stock. The time periods corresponding to these highest-scoring matches are also emphasized in a red color in the line chart. Each emphasized chart segment is interactively and bi-directionally linked with its dedicated text snippet, which describes the corresponding segment's trend label, start date, and end date. Hovering over a chart segment fades out other emphasized segments and will highlight the corresponding text in gray; hovering over a text snippet works similarly. If a stock has more than three matches, the user can expand the tile to show a list of the rest of the matches; hovering over each list item then highlights the corresponding trend in the line chart.

When results do not exactly match the user input, a notification box (Fig. 15.2) informs the user which terms are not being matched exactly. The faceting sidebar (Fig. 15.3) allows users to optionally filter the results to only include specific labels of interest. The checkbox filters are nested hierarchically by individual label families, e.g., "soaring" is a parent of both "slow soaring" and "fast soaring," based on the notion of semantic hierarchy discussed at the end of Section 3.1.3.

4.3 Semantic Parser

We implement a semantic parser module for parsing trend search queries that contain semantic labels, attributes, and temporal filter attributes. Given that the premise of SlopeSeeker is to demonstrate the utility of the quantified semantic trends dataset in the context of a search tool, we focused on supporting the interpretation of queries specifically intended to search for trends within the stock data. Prior research has demonstrated the effectiveness of a semantic parser in converting NL into a structured representation, which allows for explicit reasoning, reduced ambiguity, and consistent interpretation [59]. Semantic parsers also provide the convenience of better traceability and are performant for structured tasks. Future work could consider combining both semantic parsers for structured tasks and LLMs for open-ended tasks in the context of a more comprehensive analytics tool.

We implement our semantic parser using an open-source Python NLP library, SpaCy [29], that employs compositional semantics to identify tokens and phrases based on their semantics to create a valid parse tree from the input search query. The parser takes as input the individual tokens in the query and assigns semantic roles to these tokens. The semantic roles are one of four categories: (1) `event_type` (single or multi-sequence), (2) `trend_terms` (e.g., "tanking" or "plateau"), (3) `attr` (data attribute names such as stock ticker symbols or company names), and (4) `date_range` (absolute and relative data ranges). The tokens and their corresponding semantic roles are translated into a machine-interpretable form that can be processed to retrieve relevant search results in SlopeSeeker. For an input search query, "Show me when Alaska Airlines was tanking before November 2016," the parser output is as follows:

4.4 Search Framework

The goal of the search framework is to take the `trend_terms` tokens identified by the parser (as well as the `attr` and `date_range`, if applicable) to return relevant results. Each labeled trend event is considered an Elasticsearch "document" in our search context. Documents are the basic units stored in an Elasticsearch index. Once added to the search index, indexed documents can be first retrieved and then ranked according to a match score. We combine Elasticsearch's built-in scoring logic with our own visual saliency score to produce a scoring mechanism tailored to our use case.

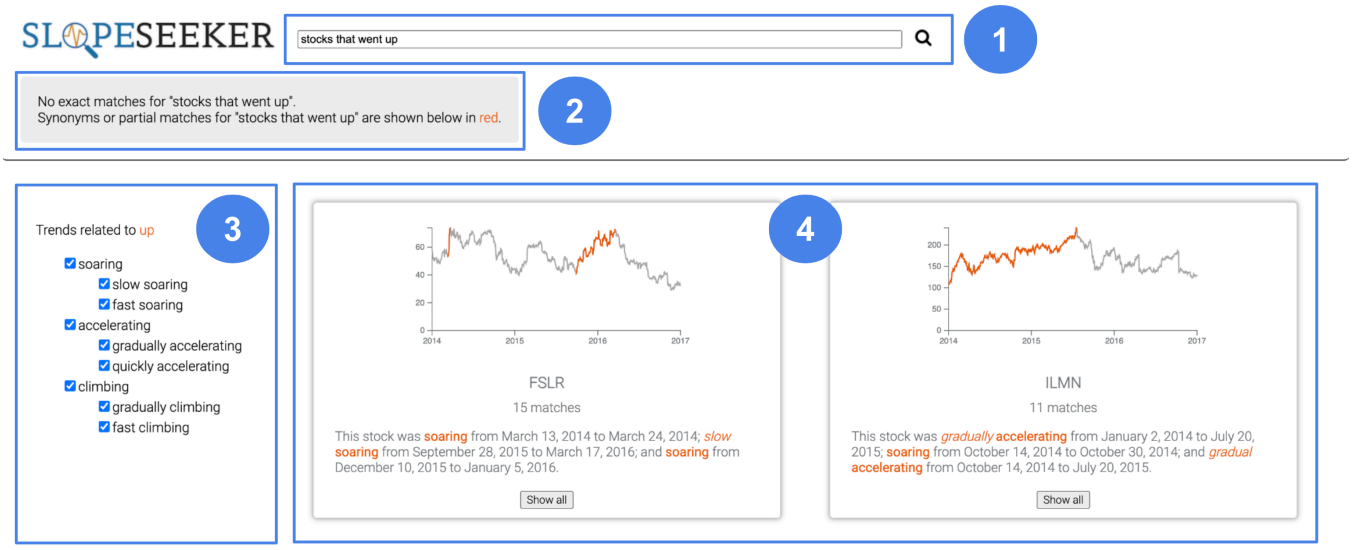

Figure 15: The SlopeSeeker interface. (1) The search box accepts natural language search queries from the user. (2) The notification box informs the user of inexact matches in the search results. (3) The faceting sidebar enables hierarchical filtering of the results. (4) Search results are shown as interactive tiles containing line charts and textual annotations.

Finally, matching documents are grouped by their parent chart for presentation to the user.

4.4.1 Indexing. The indexing phase creates indices for each document in a dataset along with their metadata. Each of the n documents (i.e., each labeled event – a portion of a line chart identified by a chart ID, start point, end point, and set of labels) is represented as a document vector d_i where:

$$\mathcal{D} = \{d_1, d_2, \cdots, d_n\}$$

We also store sets of string tokens from each document vector to support both partial and exact matches at search time:

$$\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$$

where $s_i = \varepsilon(d_i)$ for an encoding function ε that converts each document vector into a set of string tokens. The original vectors $\mathcal{D}$ and encoded tokens $\mathcal{S}$ are stored in the semantic search engine index by specifying the *mapping* of the content, which defines the type and format of the fields in the index. In other words, each semantic trend label and its associated stock data are stored as tokens in the search index in multiple processed formats (i.e., in different fields), enabling fast and flexible retrieval at search time. This indexing enables full-text search on the labels in the index, supporting exact-value search, fuzzy matching to handle typos and spelling variations, and n-grams for multi-word label matching. A scoring algorithm, tokenizers, and filters are specified as part of the search index *settings*.

4.4.2 Search (Individual Documents). The search phase can be conceptualized as having two steps – retrieval and ranking. For retrieval, consider a user input query q that is represented as a query vector $\hat{q}$ with query tokens $q_1, q_2, \cdots, q_j$. We encode $\hat{q}$ into string tokens using the same encoding function ε from indexing, such that $\hat{s} = \varepsilon(\hat{q})$. The search retrieval process then returns the most relevant r document vectors $\mathcal{R} = \{d_1, d_2, \cdots, d_r\}$ based on the degree of overlap between the set of query string tokens $\hat{s}$ and the document string tokens in $\mathcal{S}$. Specifically, the scoring function r_{max} maximizes search relevance as follows:

$$\{d_1, d_2, \cdots, d_r\} = r_{max\ i \in \{1,2,\cdots,n\}} |\hat{s} \cap s_i|$$

For search inputs that contain both a noun/verb descriptor (e.g., "decline") and a modifying adjective (e.g., "fast"), we subsequently filter out partially matching documents that contain only the adjective. This logic would prevent a query of "fast decline" from returning documents labeled "fast increase" as partial matches, for example. More formally, if $\hat{s}$ contains at least one token that matches a noun/verb descriptor in at least one document, then every matching document d_i must contain that descriptor in its set of string tokens s_i. However, users may still enter search queries consisting only of an adjective and see documents where that adjective is paired with a variety of noun/verb descriptors.

After retrieval, SlopeSeeker ranks document results based on two components. The first component is how precisely the search term matches the event labels of the document. Consider a document with a single event label. We utilize a simple scoring scheme where this document's score is the frequency with which the search terms occur in its label, divided by the length of its label, i.e., events with longer labels (e.g., those with modifying adjectives like "slow" or "fast") will be scored higher than events with shorter labels if and only if the additional tokens accounting for the added length match the search terms. (Note that only `trend_terms` parsed search tokens affect scoring, while `attr` or `date_range` tokens are simply used during retrieval to filter results.)

Consider a document d_1 with the label "slow climbing." For a search query of "slow climbing," the score for the document would

be $\frac{2}{\sqrt{12}} \approx 0.577$ since it has 2 matching tokens and 12 non-space characters in its label, and thus the label score for d_1 for this search would be 0.577. Now additionally consider document d_2 with the label "climbing" and a query concerning stocks that are "climbing." The label score for d_1 will be $\frac{1}{\sqrt{12}} \approx 0.289$ while the label score for d_2 will be $\frac{1}{\sqrt{8}} \approx 0.354$, demonstrating how longer labels with the same number of matching tokens are penalized for being less precise matches.

The second scoring component is the visual saliency score of the document's labeled event (Section 3.3), and the final composite score used to rank events in the results is then the product of the Elasticsearch and visual saliency components. The visual saliency component of scoring is most useful when there are a large number of matching results for a user query. Consider a case where the user is interested in "stocks that increased." There could feasibly be very many document results with a label of "increasing" which will all have identical (or at least very similar) Elasticsearch scores. However, these results are not likely to all be of equal interest to the user. For instance, a short three-day increase in stock price is probably less interesting, both visually and in terms of the analytical task at hand, compared to a three-month increase during which much more stock value was gained. (Note that these could both have similar slopes and thus identical labels.) The visual saliency scoring component thus serves as a tiebreaker to boost results with greater prominence and relevance over others that share identical labels.

4.4.3 Bucketing. The indexed data and result scoring are at the level of the document, where each document is an event, i.e., a labeled slope segment. Any individual chart (e.g., stock) could have multiple matching events for a query. Events within a bucket are sorted by their composite score. Buckets themselves are also scored; the final score for each bucket is the sum of the composite scores of its individual events, and buckets are presented in sorted order according to this final score. We chose this scheme to create an experience akin to standard document search, where more matches in a bucket bump that bucket higher in the results. Figure 16 demonstrates the differences between high-scoring results (buckets) for the queries "falling slowly" and "falling fast," respectively.

4.4.4 Sequence Queries. A sequence query consists of a list of trend events (single-word or multi-word) in a specified order, and the sequence query results are generated as follows. First, each individual constituent event is run through Elasticsearch as its own single-word or multi-word query but not yet bucketed. Then, sequences are constructed by taking these results and performing an SQL join based on chart identifier and start/end dates (with a tunable parameter to allow for some temporal delay between adjacent events).

We also have partial matching support for sequences. In particular, we support two types of sub-sequences: edge sub-sequences (e.g., examples for "up, flat, down" include "up" and "up, flat") as well as other in-order sub-sequences (e.g., examples for "up, flat, down" include "flat," "down," and "flat, down").

We additionally define a scoring scheme for sequences and partial sequence matches. At first, each sequence's score is assigned to be the sum of the composite scores of its constituent segments.

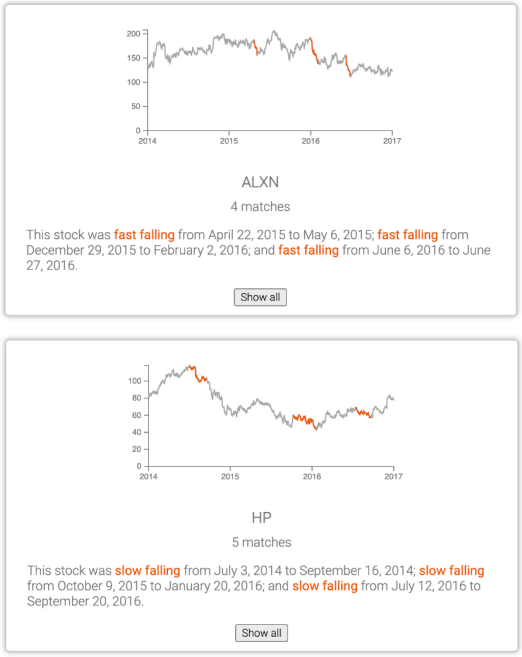

Figure 16: Users can employ modifying adjectives such as "slowly" (bottom) or "fast" (top) to provide SlopeSeeker with semantic information about the types of trends they want to see.

Although partial matching sequences have fewer constituent components and will generally have lower composite scores than full matches, we found it beneficial to additionally down-weight the composite scores of partial sequence matches. In particular, we use the following formula where the un-penalized score is notated score_0, the number of events in the sequence being scored is notated l_{seq}, the number of events in the query is notated $l_{\hat{q}}$, and the sequence offset is the number of sequential events missing from the beginning of the sequence compared to the query:

$$\text{score}_0 \left(\frac{l_{seq}}{l_{\hat{q}} + \text{offset}_{seq}} \right)^2$$

This custom scoring scheme applies two different penalties. First, longer sub-sequences ($l_{seq} \approx l_{\hat{q}}$) are penalized less and thus scored higher than shorter ones ($l_{seq} < l_{\hat{q}}$). For example, if the user is interested in stocks with the pattern "up, flat, down" which has length three, a sub-sequence of length two (e.g., "up, flat") will be scored higher than a sub-sequence of length one (e.g., "up") because the sub-sequence matches more constituent events of the sequence. Second, a non-edge sub-sequence (with a large offset) will be penalized and scored lower than an edge sub-sequence (with zero offset). Continuing with the same example, the sub-sequence "up, flat" has zero offset because it begins at the same place as the initial query pattern, but "flat, down" has an offset of one since it starts one event later in the sequence. Intuitively, sub-sequence partial matches that begin similarly to the desired sequence from

Figure 17: A top search result when the user searches for the sequence "up, down, up." Each color corresponds to a unique position in the sequence.

the query should be scored higher than those that end similarly to the desired sequence. Finally, after applying any score penalties to the sequence results and any partial results, all results are bucketed, and the bucket scores are computed as before.

5 PRELIMINARY EVALUATION OF SLOPESEEKER

Using SlopeSeeker as a design probe, we conducted a preliminary evaluation to gather feedback on the utility of the quantified semantic label dataset in the context of a search tool for trends.

5.1 Participants and Setup

We recruited 12 participants (P1-P12, six male and six female) through a mailing list at an analytics software company. Participants volunteered their time on a first-come, first-served basis, and due to company policy, they were not compensated for their participation. Based on self-reporting, participants comprised three data scientists, three sales consultants, two product managers, one program manager, one account executive, one UX researcher, and one HR analyst. Half of the participants reported that they perform data analysis on a regular basis (daily or almost daily), while the other half reported that they occasionally perform data analysis (weekly or biweekly). All participants reported that they regularly use a search tool like Google.

All sessions were conducted in person. The SlopeSeeker tool was hosted on a local server running on the experimenter's laptop[2]. The audio, video, and on-screen interactions were recorded for all sessions after receiving permission from each participant.

5.2 Procedure

Study sessions lasted about 45 minutes and followed the protocol outline below:

[~10 min.]: Participants were given an overview of the evaluation and were asked to self-report their relevant background information in visual analysis. Participants were then briefly introduced to the SlopeSeeker interface. The introduction included the capabilities of the search tool without providing any explicit NL queries to avoid biasing participants.

[~25 min.]: Participants were given a set of four tasks involving a dataset of labeled time series stock data and were asked to complete the tasks using SlopeSeeker. The first three tasks were designed to prompt users to utilize at least one of the types of supported queries, i.e., single slopes (downward and upward slopes) and multi-segment shapes. The task prompts were as follows: (1) "find an instance where a stock gained a lot of value (in a certain year or time frame)," (2) "Find an instance where a different stock lost only a small amount of value," and (3) "Find two stocks whose price followed this pattern" (showing a visual of a valley). The fourth task was open-ended, wherein participants were prompted to identify a stock they would want to invest in based on the historical patterns of stock prices in the dataset.

[~10 min.]: The sessions concluded with a post-session questionnaire (Figure 18), ten questions from the standard System Usability Scale (SUS) questionnaire [5] to help evaluate the prototype's usability, and a semi-structured interview discussing participants' overall experience using SlopeSeeker for trend search and areas for further improvement.

The SlopeSeeker tutorial, study protocol, and questionnaire are included in the supplementary material.

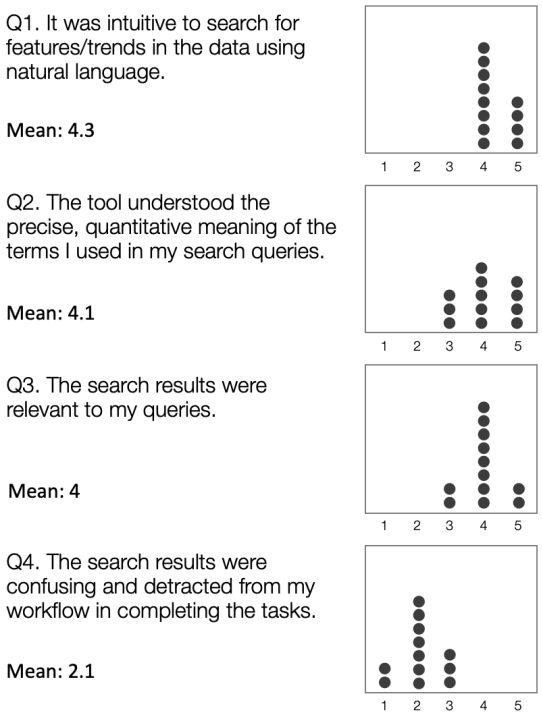

Figure 18: Participant responses to post-session questions about utterance recommendations in SlopeSeeker. Statements were rated on a scale of 1 (Strongly Disagree) to 5 (Strongly Agree).

[2]2.4 GHz MacBook Pro running macOS Ventura 13.5 set to a resolution of 3072 × 1920.

5.3 Results and Discussion

Overall, participants found SlopeSeeker to be a useful tool for searching and exploring trends in the stock data. We detail participant feedback and usage behavior with respect to the tasks and open-ended exploration during the evaluation. All participants were able to complete the four tasks, spending between 16–22 minutes (mean: 18 min.). Participants gave SlopeSeeker an average SUS score of 79.4 (a score of $\geq$ 68 is considered as an indicator of good usability [5]).

5.3.1 Intuitiveness of Interface to Search for Trends. Participants generally agreed that SlopeSeeker was intuitive for trend exploration (Figure 18, Q1). For instance, P2 found the interface and the faceted search to be similar to that of familiar search systems – "*This looks similar to Google and Amazon, and I could get started right away.*" Participants predominantly used the facet filters to see more general and specific trends such as navigating between "*slight* increase" and "increase." P7 noted the "*ability to drill down into a specific trend description*" by using the faceted search options. Participants also found the display of results in the interface to be useful. P1 commented, "*I like the tile-like view of all the trends. Helps with quick glanceability, and I know what's going on.*"

5.3.2 Interpretation of Search Queries. Participants indicated that SlopeSeeker appropriately interpreted their search queries for single slopes and multi-segment shapes (Figure 18, Q2). However, we noticed that the quality of search results deteriorated when queries included additional information such as relative time periods (e.g., "did the apple stock go up *recently*?"), subjective concepts (e.g., "*best* stocks to buy in 2014"), or attributes that were not included in our stock dataset. However, all participants appreciated the text accompanying each trend result that showed the quantifiable trend with the corresponding trend segment in the chart upon hover (Figure 15.4), commenting, "*this is pretty neat*" [P5] and "*I can compare the labels for what 'cliff' means and see it visually. That's helpful to know what's going on*" [P11].

5.3.3 Relevance of Search Results. Generally, participants were in agreement that the search results were relevant to their input queries (Figure 18, Q3 and Q4). Participants specifically appreciated that the tool could differentiate between nuances in quantifiable semantics in the trends. P7 cited an example from her session and said, "*Wow, I asked for trends that hit a cliff in 2014, and it's pretty cool to see an array of results with cliffs in them.*" Others found SlopeSeeker's capability to detect a sequence of trends to be useful. P12 stated, "*I was curious to see what stocks went up and then suddenly down, and I was impressed that it recognized 'suddenly' for 'down.'*" However, there were also some mixed reactions on the relevance of the results, where participants indicated limitations in the capabilities of the tool, as described in the takeaways below:

Consider pragmatics in search. The evaluation indicated a need to consider pragmatics to support a more natural conversational flow in search interfaces, a paradigm present in various natural language interfaces for data exploration [44, 49]. Several (5 out of 12) participants typed a full query such as, "*which stocks went up in 2014?*," followed by an underspecified query, "*what about 2015?*" that would need to be interpreted in the context of the previous search history. P3 commented, "*I'm now used to just asking the next question assuming the system already knows what I mean. I expect the same [behavior] here too.*"

Integrate trend search with visual analytical tools. For supporting a more comprehensive data exploration, SlopeSeeker would need to be integrated into visual analysis tools that support a wider range of analytical inquiry. For example, three participants wanted to see categories of stock that did poorly or compare their relative performance with each other, as noted by P1: "*while I saw a bunch of stock tanking around the same time, I'd like to bucket them into a multi-line chart by tech vs. retail stock to get a better understanding of the trend patterns.*"

Provide external knowledge that gives context around a certain trend pattern. A deeper understanding of any trend necessitates an awareness of the contextual information surrounding it. While the search results in SlopeSeeker indicated a certain trend pattern, external knowledge can provide the *why* and *how* behind such a pattern, such as augmenting data with external information from knowledge graphs and web corpora [12]. 8 out of 12 participants expressed a need for including additional context beyond what is in the underlying data to enable them to make informed decisions and assess how the trends are influenced by external events. P4 stated, "*I can see what is the trend, but I'm curious to know more. Why did that stock suddenly go bust at that time? Feels like I want a Google button right next to that sharp drop.*"

Support control over time granularity as well as fuzzy time concepts. Finally, SlopeSeeker returns trend results by year, and participants expressed the need for more flexibility in exploring trends with varying granularity, from specific dates to weekly, monthly, and quarterly views, as well as fuzzy temporal descriptors, like "stocks that remained stagnant for an extended period." P5 wanted to see the "*fluctuating trend quarter-by-quarter to check if there's anything seasonal going on there.*" Computing trend patterns at different levels of time granularity could support more nuanced analyses and expose patterns such as seasonal variations or temporal outliers.

6 FUTURE DIRECTIONS FOR SEMANTIC TRENDS DATASET AND ITS APPLICATIONS

In this work, we demonstrate the utility of a labeled dataset of semantic trends and their properties by implementing SlopeSeeker to help search and discover trends in line charts. However, we envision multiple directions for future work to utilize our dataset and contributions to go beyond the current capabilities of SlopeSeeker.

6.1 Extending Semantic Trends Dataset

One future direction for extending our dataset is to support users searching for more global descriptions of time series data behavior over a longer period of time (e.g., a user may want to search for when a certain stock was "volatile" versus "consistent"). We could also leverage the fact that a conceptual duality exists between event sequences and global descriptors. For instance, one way of expressing that a stock is "volatile" is to describe the price as going up and down repeatedly, which could be represented by a sequence of trend events comprising "up" and "down" events.

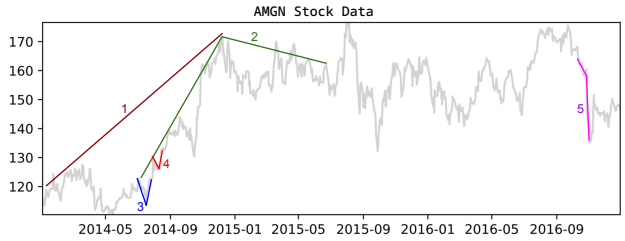

Figure 19: GPT-generated narrative based on quantitative semantic observations. Line color is for clarity only. (Numbers) refer to trend identifiers. *"In the realm of biotechnology stocks, AMGN exhibited a distinct pattern of ups and downs. The stock started 2014 by (1) slowly expanding, growing by 32% until December of that year. This upward movement evolved into a (2) hump shape that continued until July 2015, marking a 20% increase from its starting point in mid-2014. The larger events were interspersed with smaller but still noteworthy fluctuations. For instance, the stock experienced a (3) rebound in July 2014, followed by a (4) lull that extended into September. Towards the end of 2016, the stock went through a (5) cliff-like decline of 17%, a sharp contrast to its earlier growth. These smaller events added intricate layers to AMGN's overall performance, rendering it a stock of compelling dynamics."*

LLMs, when trained on domain-specific corpora, can also help identify and label trends that are specifically meaningful to a particular domain. For instance, rather than identifying a trend as simply "positive" or "negative," LLMs could be used to discern and label subtleties such as "bullish," "stagnant," and "bearish" for market trends, "growth," "plateau," and "decline" for financial data, or "increasing," "steady," and "falling" for weather information. LLMs also show promise in labeling trends at different temporal scales based on the data domain. We hence think it would be worthwhile to explore leveraging LLMs to label trends across different temporal scales – for example, a short-term "daily rally," a medium-term "monthly correction," or a long-term "annual growth trajectory."

While our work introduces a search tool for trends, there are still opportunities with respect to data presentation (e.g., text interfaces, text summaries, accessibility tools, etc.). LLMs could be employed to craft a narrative from a set of quantitative semantic observations such as those depicted in Figures 9, 10, and 11. For example, in Figure 19, we asked GPT-4 to respect the specific trend labels but, if appropriate, to aggregate several events with labels like "fluctuation." This ability to "smooth out" a set of events into a single engaging narrative could provide alternative ways to consume quantitative semantic observations. However, given the possibility of LLMs hallucinating in their responses, future work should also consider methods of checking for hallucinations in LLM-generated narratives and exposition text [63].

6.2 Future Extensions of Trend Search Tools

Our work also points to potential new modalities and approaches for trend querying. For instance, the underlying labeled dataset and subsequent search techniques can be integrated with sketch-based input to help bridge the semantic gap between visual specification and trend semantics. While SlopeSeeker incorporates domain-specific labels for describing trends in data, the labeled dataset could be applied to provide analysis and data narratives bespoke to that domain. Search tools and visual analytics tools can also employ distinct lexicons to provide more targeted data exploration, insight, and narrative generation. While SlopeSeeker currently supports search queries that involve trend descriptors and modifiers such as "fast falling" and "slow rising," the tool can be augmented with additional corpora and knowledge, including LLMs, to offer interpretations of trends that are contextually relevant, such as the query, "*Do stocks always fall after they bounce twice?*".

7 CONCLUSION

Search systems have begun to support basic analytical intents when displaying data and charts in response to users' natural language queries. However, user workflows often expect more specific tasks than can be robustly handled by search tools, such as identifying relevant trends in temporal data. In this paper, we present a dataset of trend descriptor labels and associated quantified semantics and then employ this dataset in a search tool called SlopeSeeker, which supports diverse trend search intents. The tool utilizes custom scoring and ranking logic to return relevant results based on users' natural language queries. A preliminary evaluation of SlopeSeeker demonstrates that the tool is intuitive for finding trends in data, and the underlying quantifiable semantic trend labels provide relevant search results for various nuances of trend descriptors in the input queries. We hope that our publicly available semantic trend label dataset can enable future research in developing intelligent search interfaces that can understand and leverage precise quantified semantics and support users' increasingly diverse visual data analysis intents.

REFERENCES

[1] 2023. Elasticsearch. https://www.elastic.co/elasticsearch/.
[2] 2023. Flask. https://flask.palletsprojects.com/en/3.0.x/.
[3] 2023. IBM Watson Analytics. http://www.ibm.com/analytics/watson-analytics.
[4] 2023. React. https://react.dev/.
[5] 2023. System Usability Scale (SUS). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.
[6] 2023. Tableau Software. https://www.tableau.com.
[7] 2023. ThoughtSpot. http://www.thoughtspot.com.
[8] Eija Airio, Kalervo Järvelin, Pirkko Saatsi, Jaana Kekäläinen, and Sari Suomela. 2004. *CIRI - An Ontology-based Query Interface for Text Retrieval* (1 ed.). Number 20 in Publications of the Finnish Artificial Intelligence Society. Finnish Artificial Intelligence Society, 73–82.
[9] Robert Amar, James Eagan, and John Stasko. 2005. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, 111–117. https://doi.org/10.1109/INFVIS.2005.1532136
[10] Dennis Bromley and Vidya Setlur. 2023. What Is the Difference Between a Mountain and a Molehill? Quantifying Semantic Labeling of Visual Features in Line Charts. *IEEE Transactions on Visualization and Computer Graphics* (2023).
[11] D. Buscaldi, Paolo Rosso, and Emilio Sanchis Arnal. 2005. A WordNet-based Query Expansion Method for Geographical Information Retrieval. In *Conference and Labs of the Evaluation Forum.*
[12] Dylan Cashman, Shenyu Xu, Subhajit Das, Florian Heimerl, Cong Liu, Shah Rukh Humayoun, Michael Gleicher, Alex Endert, and Remco Chang. 2021. CAVA: A Visual Analytics System for Exploratory Columnar Data Augmentation Using Knowledge Graphs. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1731–1741. https://doi.org/10.1109/TVCG.2020.3030443
[13] Chris Chatfield. 2004. *The Analysis of Time Series: An Introduction* (6th ed.). CRC Press, Florida, US.
[14] Gong Cheng, Weiyi Ge, and Yuzhong Qu. 2008. Falcons: Searching and Browsing Entities on the Semantic Web. In *Proceedings of the 17th International Conference*

on World Wide Web (Beijing, China) *(WWW '08)*. Association for Computing Machinery, New York, NY, USA, 1101–1102. https://doi.org/10.1145/1367497.1367676

[15] Philipp Cimiano, Peter Haase, Jörg Heizmann, Matthias Mantel, and Rudi Studer. 2008. Towards Portable Natural Language Interfaces to Knowledge Bases - The Case of the ORAKEL System. *Data Knowl. Eng.* 65, 2 (May 2008), 325–354. https://doi.org/10.1016/j.datak.2007.10.007

[16] Olivier Corby, Rose Dieng-Kuntz, and Catherine Faron-Zucker. 2004. Querying the Semantic Web with the CORESE search engine. *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, 705–709.

[17] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. 2010. Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In *Extended Semantic Web Conference*.

[18] MC Ferreira De Oliveira and Haim Levkowitz. 2003. From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 9, 3 (2003), 378–394.

[19] Kedar Dhamdhere, Kevin S. McCurley, Ralfi Nahmias, Mukund Sundararajan, and Qiqi Yan. 2017. Analyza: Exploring Data with Conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI 2017)*. 493–504.

[20] David H Douglas and Thomas K Peucker. 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122.

[21] Miriam Fernandez, Vanessa Lopez, Marta Sabou, Victoria Uren, David Vallet, Enrico Motta, and Pablo Castells. 2008. Semantic Search Meets the Web. In *2008 IEEE International Conference on Semantic Computing*. 253–260. https://doi.org/10.1109/ICSC.2008.52

[22] Leo Ferres, Avi Parush, Zhihong Li, Yandu Oppacher, and Gitte Lindgaard. 2006. Representing and Querying Line Graphs in Natural Language: The *iGraph* System. In *Smart Graphics, 6th International Symposium, SG 2006, Vancouver, Canada, July 23-25, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4073)*, Andreas Butz, Brian D. Fisher, Antonio Krüger, and Patrick Olivier (Eds.). Springer, 248–253. https://doi.org/10.1007/11795018_25

[23] Tim Finin, Li Ding, Rong Pan, Anupam Joshi, Pranam Kolari, Akshay Java, and Yun Peng. 2005. Swoogle: Searching for Knowledge on the Semantic Web. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 4* (Pittsburgh, Pennsylvania) *(AAAI'05)*. AAAI Press, 1682–1683.

[24] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. 2015. DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology (UIST 2015)*. ACM, New York, NY, USA, 489–500.

[25] Thomas R. Gruber. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 2 (1993), 199–220. https://doi.org/10.1006/knac.1993.1008

[26] A. Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Seán O'Riain, and Stefan Decker. 2007. SWSE: Answers Before Links!. In *Semantic Web Challenge*.

[27] Jeff Heflin, James A. Hendler, and Sean Luke. 2003. SHOE: A Blueprint for the Semantic Web. In *Spinning the Semantic Web*.

[28] Harry Hochheiser and Ben Shneiderman. 2004. Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration. *Information Visualization* 3 (2004), 1 – 18. https://api.semanticscholar.org/CorpusID:5628923

[29] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. (2020). https://doi.org/10.5281/zenodo.1212303

[30] E. Hoque, P. Kavehzadeh, and A. Masry. 2022. Chart Question Answering: State of the Art and Future Directions. *Computer Graphics Forum* 41, 3 (2022), 555–572. https://doi.org/10.1111/cgf.14573

[31] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2017. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2017), 309–318.

[32] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. 2006. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. (01 2006).

[33] Dae Hyun Kim, Vidya Setlur, and Maneesh Agrawala. 2021. Towards Understanding How Readers Integrate Charts and Captions: A Case Study with Line Charts. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 610, 11 pages. https://doi.org/10.1145/3411764.3445443

[34] Robert Kincaid and Heidi Lam. 2006. Line Graph Explorer: Scalable Display of Line Graphs Using Focus+Context. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. 404–411.

[35] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sapporo, Japan, 423–430. https://doi.org/10.3115/1075096.1075150

[36] Doris Jung-Lin Lee, John Lee, Tarique Siddiqui, Jaewoo Kim, Karrie Karahalios, and Aditya Parameswaran. 2020. You Can't Always Sketch What You Want: Understanding Sensemaking in Visual Query Systems. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 1267–1277. https://doi.org/10.1109/TVCG.2019.2934666

[37] Yuangui Lei, Victoria S. Uren, and Enrico Motta. 2006. SemSearch: A Search Engine for the Semantic Web. In *International Conference Knowledge Engineering and Knowledge Management*.

[38] V. López, Michele Pasin, and Enrico Motta. 2005. AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. In *Extended Semantic Web Conference*.

[39] Vanessa Lopez, Marta Sabou, and Enrico Motta. 2006. PowerMap: Mapping the Real Semantic Web on the Fly. In *Proceedings of the 5th International Conference on The Semantic Web* (Athens, GA) *(ISWC'06)*. Springer-Verlag, Berlin, Heidelberg, 414–427. https://doi.org/10.1007/11926078_30

[40] Microsoft. 2023. Microsoft PowerBI. https://powerbi.microsoft.com/.

[41] George A Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[42] Fábio Miranda, Marcos Lage, Harish Doraiswamy, Charlie Mydlarz, Justin Salamon, Yitzchak David Lockerman, Juliana Freire, and Cláudio T. Silva. 2018. Time Lattice: A Data Structure for the Interactive Visual Analysis of Large Time Series. *Computer Graphics Forum* 37 (2018). https://api.semanticscholar.org/CorpusID:51873868

[43] Dan I. Moldovan and Rada Mihalcea. 2000. Using WordNet and Lexical Operators to Improve Internet Searches. *IEEE Internet Computing* 4, 1 (jan 2000), 34–43. https://doi.org/10.1109/4236.815847

[44] Arpit Narechania, Arjun Srinivasan, and John Stasko. 2021. NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 369–379. https://doi.org/10.1109/TVCG.2020.3030378

[45] OpenAI. 2023. GPT-4 technical report. *arXiv* (2023), 2303–08774.

[46] Eyal Oren, Christophe Guéret, and Stefan Schlobach. 2008. Anytime Query Answering in RDF through Evolutionary Algorithms. In *Proceedings of the 7th International Conference on The Semantic Web* (Karlsruhe, Germany) *(ISWC '08)*. Springer-Verlag, Berlin, Heidelberg, 98–113. https://doi.org/10.1007/978-3-540-88564-1_7

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[48] PostgreSQL Global Development Group. 2023. PostgreSQL. https://www.postgresql.org/

[49] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. 2016. Eviza: A Natural Language Interface for Visual Analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST 2016)*. ACM, New York, NY, USA, 365–377.

[50] Vidya Setlur, Andriy Kanyuka, and Arjun Srinivasan. 2023. Olio: A Semantic Search Interface for Data Repositories. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software Technology* (San Francisco, California) *(UIST 2023)*. ACM, New York, NY, USA. https://doi.org/10.1145/3586183.3606806

[51] Vidya Setlur, Melanie Tory, and Alex Djalali. 2019. Inferencing Underspecified Natural Language Utterances in Visual Analysis. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Rey, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 40–51. https://doi.org/10.1145/3301275.3302270

[52] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2016. Effortless Data Exploration with Zenvisage: An Expressive and Interactive Visual Analytics System. *Proc. VLDB Endow.* 10, 4 (nov 2016), 457–468. https://doi.org/10.14778/3025111.3025126

[53] Tarique Siddiqui, Paul Luh, Zesheng Wang, Karrie Karahalios, and Aditya G. Parameswaran. 2021. From Sketching to Natural Language: Expressive Visual Querying for Accelerating Insight. *SIGMOD Rec.* 50, 1 (jun 2021), 51–58. https://doi.org/10.1145/3471485.3471498

[54] Tom Soukup and Ian Davidson. 2002. *Visual Data Mining: Techniques and Tools for Data Visualization and Mining*. John Wiley & Sons.

[55] Rohini K. Srihari and W. Li. 1999. Information Extraction Supported Question Answering. In *Text Retrieval Conference*.

[56] Arjun Srinivasan and John Stasko. 2018. Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 511–521.

[57] Granville Tunnicliffe Wilson. 2016. Time Series Analysis: Forecasting and Control,5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1. *Journal of Time Series Analysis* 37 (03 2016). https://doi.org/10.1111/jtsa.12194

[58] Chong Wang, Miao Xiong, Qi Zhou, and Yong Yu. 2007. PANTO: A Portable Natural Language Interface to Ontologies. In *Proceedings of the 4th European Conference on The Semantic Web: Research and Applications* (Innsbruck, Austria) *(ESWC '07)*. Springer-Verlag, Berlin, Heidelberg, 473–487. https://doi.org/10.1007/978-3-540-72667-8_34

[59] Yuk Wah Wong and Raymond Mooney. 2006. Learning for Semantic Parsing with Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. Association for Computational Linguistics, New York City, USA, 439–446. https://aclanthology.org/N06-1056

[60] B. Yu and C. T. Silva. 2020. FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE Transactions on Visualization and Computer Graphics* 26, 01 (jan 2020), 1–11. https://doi.org/10.1109/TVCG.2019.2934668

[61] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2009. From Keywords to Semantic Queries-Incremental Query Construction on the Semantic Web. *Web Semant.* 7, 3 (Sep 2009), 166–176. https://doi.org/10.1016/j.websem.2009.07.005

[62] Yue Zhao, Yunhai Wang, Jian Zhang, Chi-Wing Fu, Mingliang Xu, and Dominik Moritz. 2022. KD-Box: Line-segment-based KD-tree for Interactive Exploration of Large-scale Time-Series Data. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 890–900. https://doi.org/10.1109/TVCG.2021.3114865

[63] Guido Zuccon, Bevan Koopman, and Razia Shaik. 2023. ChatGPT Hallucinates when Attributing Answers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (Beijing, China) *(SIGIR-AP '23)*. Association for Computing Machinery, New York, NY, USA, 46–51. https://doi.org/10.1145/3624918.3625329