

# VIANA: Visual Interactive Annotation of Argumentation

Fabian Sperrle\*

Rita Sevastjanova\*

Rebecca Kehlbeck\*

Mennatallah El-Assady\*

University of Konstanz

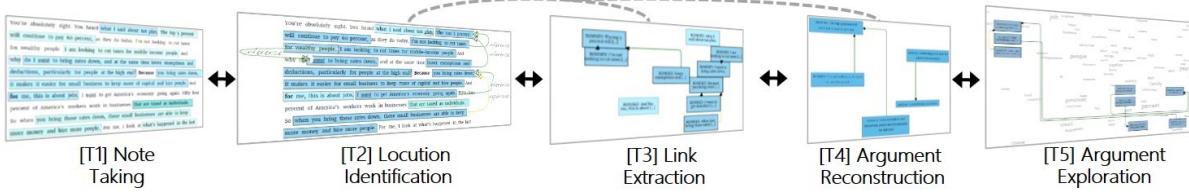


Figure 1: VIANA is a system for interactive annotation of argumentation. It offers five different analysis layers, each represented by a different view and tailored to a specific task. The layers are connected with semantic transitions. With increasing progress of the analysis, users can abstract away from the text representation and seamlessly transition towards distant reading interfaces.

## ABSTRACT

Argumentation Mining addresses the challenging tasks of identifying boundaries of argumentative text fragments and extracting their relationships. Fully automated solutions do not reach satisfactory accuracy due to their insufficient incorporation of semantics and domain knowledge. Therefore, experts currently rely on time-consuming manual annotations. In this paper, we present a visual analytics system that augments the manual annotation process by automatically suggesting which text fragments to annotate next. The accuracy of those suggestions is improved over time by incorporating linguistic knowledge and language modeling to learn a measure of argument similarity from user interactions. Based on a long-term collaboration with domain experts, we identify and model five high-level analysis tasks. We enable close reading and note-taking, annotation of arguments, argument reconstruction, extraction of argument relations, and exploration of argument graphs. To avoid context switches, we transition between all views through seamless morphing, visually anchoring all text- and graph-based layers. We evaluate our system with a two-stage expert user study based on a corpus of presidential debates. The results show that experts prefer our system over existing solutions due to the speedup provided by the automatic suggestions and the tight integration between text and graph views.

**Keywords:** Argumentation annotation, machine learning, user interaction, layered interfaces, semantic transitions

## 1 INTRODUCTION

Argument mining is a flourishing research area that enables various novel, linguistically-informed applications like semantic search engines, chatbots or human-like discussion systems, as convincingly demonstrated by IBM’s project debater [58]. To achieve reliable performance in these complex tasks, modern systems rely on the analysis of the underlying linguistic structures that characterize successful argumentation, rhetoric, and persuasion. Consequently, to distill the building blocks of argumentation from a text corpus, it is not sufficient to employ off-the-shelf Natural Language Processing techniques [65], which are typically developed for coarser analytical tasks (see [42] for an overview), such as with the high-level tasks of topic modeling [19] or sentiment analysis [5].

\*e-mail: firstname.lastname@uni-konstanz.de

Hence, to master the challenge of identifying argumentative substructures in large text corpora, computational linguistic researchers are actively developing techniques for the extraction of argumentative fragments of text and the relations between them [41]. To develop and train these complex, tailored systems, experts rely on large corpora of annotated gold-standard training data. However, these training corpora are difficult and expensive to produce as they extensively rely on the fine-grained manual annotation of argumentative structures. An additional barrier to unifying and streamlining this annotation process and, in turn, the generation of gold-standard corpora is the subjectivity of the task. A reported agreement with Cohen’s  $\kappa$  [11] of 0.610 [64] between human annotators is considered “substantial” [38] and is the state-of-the-art in the field. However, for the development of automated techniques, we have to rely on the extraction of decisive features. Cabrio et al. [9] present a mapping between *discourse indicators* and *argumentation schemes*, indicating a promising direction for more automation. We use such automatically extracted discourse indicators as a reliable foundation for annotation-guidance. The resulting visual analytics workflow is presented in Fig. 2. After discourse units have been annotated with discourse indicators and enriched with sentence-embedding vectors, they are used to train a measure of argument similarity. This measure is updated over time as users annotate more text. To speed up training and remove clutter from the visual interface we introduce a *novel, viewport-dependent approach to suggestion decay*.

Including machine learning and visual analytics into annotation-systems presents a substantial step towards semi-automated argumentation annotation. Systems can rely on a bi-directional learning loop to improve performance: first, they can learn from the available input data, providing both better recommendations and guidance for users. Second, systems can also learn from user interactions to improve and guide the used machine learning algorithms. Tackling these challenges and employing such progressive mixed-initiative learning, we present a novel visual analytics approach for argumentation annotation in this paper. We base our design choices on a long-term collaboration with experts from the humanities and social sciences. We observed their work processes and underlying theories to gain insight into their respective fields [29]. The well-established Inference Anchoring Theory [8] (IAT) provides the solid foundation of a theoretical framework that is capable of representing argumentative processes. Having acquired direct insight into argumentation from our collaborations, we present a requirement and task analysis that informs the development of VIANA, our annotation system, as well as future approaches in Fig. 3.

**Contributions –** While we present VIANA in the context of the

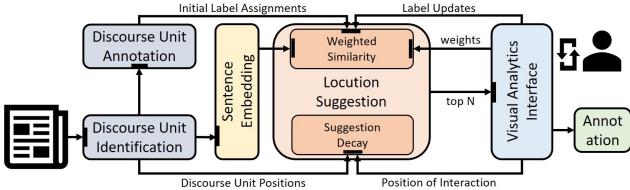


Figure 2: The locution suggestion workflow combines linguistic annotations with a language model and learns from presence and absence of user interactions to improve suggestion quality over time.

*Inference Anchoring Theory* in this paper, its concepts are readily adaptable to other domain-specific theories and annotation problems, for example from linguistics or political sciences. Thus, this paper’s contribution is two-fold. (i) We contribute a **requirement-and task-analysis** for effectively developing visual analytics systems in the field of argumentation annotation. (ii) We further contribute the **visual analytics application**, VIANA, including a **novel design of layered visual abstractions** for a targeted analysis through semantic transitions, as well as a **recommendation system** learning from both domain knowledge and user interaction, introducing **viewport-dependent suggestion decay**.

## 2 RELATED WORK

Recent years have seen a rise of interactive machine learning [22] and such techniques are now commonly integrated into visual analytics systems, as recently surveyed by Endert et al. [21]. Often, they are used to learn model refinements from user interaction [18] or provide *semantic interactions* [20]. Semantic interactions are typically performed with the intent of refining or steering a machine-learning model. In VIANA, expert users perform *implicit* semantic interactions, as their primary goal is the annotation of argumentation. The result is a concealed machine teaching process [56] that is not an end in itself, but a “by-product” of the annotation.

**Close Reading and Annotation Interfaces** – In their survey, Jänicke et al. [32] present an overview of visualization techniques which support close and distant reading tasks. According to the authors, “close reading retains the ability to read the source text without dissolving its structure.” [32] Distant reading generalizes or abstracts the text by presenting it using global features.

Several systems combine the close and distant reading metaphors to provide deeper insights into textual data, such as *LeadLine* [13] or *EMDialog* [30]. Koch et al. [36] have developed a tool called *VarifocalReader*, which combines focus- and context-techniques to support the analysis of large text documents. The tool enables exploration of text through novel navigation methods and allows the extraction of entities and other concepts. *VarifocalReader* places all close and distant-reading views next to each other, following the *SmoothScroll* metaphor by Wörner and Ertl [68]. VIANA instead “stacks” the different views into task-dependent layers.

In recent years, several web-based interfaces have been created to support users in various text annotation tasks. For example, *BRAT* [61] can be used for the annotation of POS tags or named entities. In this interface, annotations are made directly in the text by dragging the mouse over multiple words or clicking on a single word. VIANA employs the same interactions for text annotation. Another web-based annotation tool is called *Anafora* [10]; it allows annotations of named entities and their relations. Lu et al. [45] use automatic entity extraction for annotating relationships between media streams. *TimeLineCurator* [23] automatically extracts temporal events from unstructured text data and enables users to curate them in a visual, annotated timeline. Bontcheva et al. [6] have presented a collaborative text annotation framework and emphasize the importance of pre-annotation to significantly reduce annotation costs. Skeppstedt et al. have presented a framework that creates BRAT-

compatible pre-annotations [57] and discuss (dis-)advantages of pre-annotation. The initial suggestions of VIANA could be seen as pre-annotations, but are automatically updated after each interaction.

**Argument Annotation** – Scheuer et al. [54] offer a comprehensive overview of computer-supported argumentation systems. They characterize five visual argument representations, including graph views, and focus on both systems that allow students to practice the rules of argumentation and those that incorporate collaboration.

*Araucaria* [50] and its more recent online variant OVA+ [33] support the interactive diagramming of argument structures. OVA+, the de-facto standard for argumentation annotation, offers a text view and a graph view side by side. It supports a vast set of argumentation theories and their peculiarities. When annotating, users create detailed argument graphs (see Fig. 3) through text selection and align them with drag and drop, or rely on a rudimentary automatic layout engine. *Monkeypuzzle* [14] relies on the user interface and interactions introduced in *Araucaria*, but adds the possibility to simultaneously annotate texts from multiple sources. Like VIANA, both OVA and Araucaria enable annotation according to IAT. However, VIANA automatically aligns extracted argumentation graphs with the text view and provides automatically updating suggestions to speed up the annotation process. Stab et al. [60] have created a web-based annotation tool, combining a text and graph view side by side. Users create arguments directly in the transcript, and each component is assigned an individual color. Relations between arguments (attack, support, sequence) are shown in a simple graph structure. All introduced systems offer graph and text views and suffer from similar issues. It is usually hard to relate the graph structure to the original text, and large input corpora make the presented information hard to manage. VIANA tackles these issues with task-specific interface layers and seamless transitions between text- and graph views.

**Interactive Recommender Systems** – There are generally three approaches to recommender systems: collaborative filtering, content-based, and hybrid approaches [26]. Collaborative filtering systems utilize ratings or interactions from other users to recommend items [28, 40, 53], while content-based systems [44] make predictions purely on attributes of the items under consideration. Hybrid approaches combine both methods. Annotation suggestions by VIANA are content-based. Various approaches have been developed to react to changing ratings [37, 69] and evolving user preference over time [24]. Feedback to recommender systems is either explicit, for example in the form of ratings, or implicit, like in the number of times a song has been played or skipped [47] or how often an item has been preferred over another [39, 51]. Jannach et al. recently surveyed implicit feedback in recommender systems [34]. VIANA accepts both explicit (acceptance or rejection of a suggestion) and implicit feedback. To incorporate implicit feedback we propose a novel approach to recommendation decay. Based on the current viewport VIANA penalizes those suggestions that are visible, but ignored. Penalties increase with decreasing pixel distance between the suggestion and a user interaction. Previous work in recommender systems has often focussed on temporal influence decay [31, 43] to lower to influence of older actions on current recommendations.

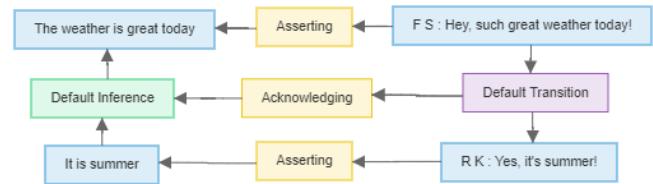


Figure 3: Simple example of an IAT (Inference Anchoring Theory [8]) structure as produced by OVA+ [33].

### 3 BACKGROUND: ARGUMENTATION ANNOTATION

The study of argumentation in political discourse has a history that spans over 2000 years. Since the foundational theories of Aristotle [3], scholars have been studying the building blocks of successful argumentation and methods of persuasion. This research evolved from a mostly theoretical disputation to the thriving field of data-driven, computational argumentation mining. In a review of the landmark book “Argumentation Machines: New Frontiers in Argumentation and Computation” [49], Zukerman defines argumentation as the “*study of different aspects of (human) interactions whose objective is to reach a conclusion about the truth of a proposition or the adoption of a course of action.*” [70] To introduce the specific terminology of the field, we provide a simplified example usage of our system. The expert user’s goal is the extraction of argumentative structures and their relations; data that is typically presented in Inference Anchoring Theory (IAT) graphs, as shown in Fig. 3. To gather the underlying data, she begins by close reading the text and annotating fragments (called *locutions*) of text. In the example, the text is a short discussion between *FS* and *RK* about the weather. The extracted locutions form the right-hand side of the graph and are connected with a *transition*, showing their logical connection. *VIANA* creates those transitions automatically based on the temporal order of locutions. Typically, locutions have exactly one associated *proposition* (left-hand side of the graph) that is also automatically created by the system. Propositions include, for example, premises and conclusions and form the building blocks of arguments. Having identified locutions, the expert identifies a support relation—called *inference*—between the associated propositions. Alternative types of interpropositional relations are attacks, also called *conflict*, and *rephrases*. Next, the expert user *reconstructs* the propositions. Reconstruction entails, for example, correction of grammatical issues caused by the extraction of text fragments from their context or capitalization. Consequently, the propositional content differs slightly from the locution. The yellow boxes in Fig. 3 are called *illocutionary connectors* and form a cornerstone of the IAT. While the connections are automatically created with their associated left-hand sides of the graph, the user proceeds to select the labels from a wide array, including “asserting”, “questioning” or “challenging”. Having completed the annotation she proceeds to an overview map showing the different concepts and topics contained in the propositions to validate her results.

Apart from the simplistic example presented above, *VIANA* can be used to annotate more complex IAT scenarios that can only be introduced very briefly here due to space limitations; the supplementary material provides more detailed explanations. *Linked Arguments* describe propositions that can only create an interpropositional relation together and not on their own. Instead of attacking a proposition, *undercuts* attack inter-propositional relations. *Indexicals* [7] are locutions like “Of course not!” that rely on the content of the previous locution and lose their meaning when separated. Further, IAT resolves reported speech into artificial locutions.

**Requirement and Task Analysis—** From our long-term collaboration with experts in philosophy and computational linguistics, we identify several requirements for systems tailored to the task of text annotation and, in particular, argumentation annotation. We categorize our collection of requirements into *general* needs and items that are *specific* to the domain of argumentation mining.

In **general**, text annotation tools should include *close and distant reading interfaces* to provide ways to work on the text while also allowing to abstract and generate higher-level insights. These interfaces need to be connected in such a way that users can easily switch between them and *avoid unnecessary losses of context*. This includes keeping the interface clean and easy to use, *removing clutter and distractions*. *User guidance* can greatly speed up the analysis process and facilitate the *curation of results* and their exploration. Once users have compiled results or insights, tools should offer

ways to *export and share* those results using *visualizations* suitable for communicating the findings to both experts and non-experts.

Systems for an efficient **argumentation annotation** need to [R1] deal with large amounts of text and [R2] extract graph structures from that text. Experts’ requirements for such systems further include the possibility to [R3] extract argumentative fragments of text as locutions and [R4] reconstruct propositions from them. Once propositions have been extracted, they also need to be able to [R5] connect propositions with relations like inference and conflict and [R6] capture argumentation schemes and annotate illocutionary forces. Furthermore, discussions often revisit previously mentioned topics, necessitating appropriate functionality to [R7] connect propositions with large temporal gaps.

From this requirement analysis we derive five abstract, high-level tasks for argumentation annotation, tailored to expert annotators and analysts: [T1] Close Reading and Note-Taking, [T2] Text Segmentation and Locution Identification, [T3] Relationship Extraction, [T4] Argumentation Reconstruction, [T5] Argument Exploration. These tasks need to be supported by systems catering to argumentation annotation. For the design and implementation of *VIANA*, we translate the five tasks to five distinct, interactive views that support completing them: the *Note Taking*, *Locution Identification*, *Link Extraction*, *Argument Reconstruction*, and *Argument Exploration* views from Fig. 1 will be introduced in detail in Sect. 4. All views are presented as stacked layers and connected via semantic transitions. With advancing annotation progress, users transition through the layers, increasing the interface abstraction and transitioning from a pure text-based view to a high-level graph abstraction.

### 4 WORKSPACE DESIGN CONSIDERATIONS

As introduced in the previous section, our system is grounded in linguistic argumentation theory. While the number of existing annotation systems conforming to the theory is limited, domain experts actively use those available tools. We thus anchored our design decisions in those accepted applications, as they can represent the linguistic theory and have already formed the experts’ mental models that are not easily changed now, as our expert user study confirmed.

From the previous long-term collaboration with said experts, we also gathered several issues with available annotation systems. One frequently mentioned complaint was the lack of a connection between extracted locutions and their context in the original transcript. We overcome this limitation by directly annotating locutions in the transcript by highlighting the respective words. To emphasize the “hand-made” nature of the annotation, we offer the option to employ sketch-rendering techniques when displaying locution annotations as well as the connections between them to encourage users to keep refining them. While we initially considered mapping the roughness of the sketch to the uncertainty of the annotation we rejected this idea as comparing different levels of “sketchiness” is extremely difficult. Wood et al. [67] compared “normal” and sketchy visualizations for different use cases. They conclude that user engagement increases with sketchy visualizations when compared to non-sketchy ones. Additionally, they note that the overall interaction with a tool is perceived as more positive if it uses sketchy rendering. Our study did, however, not fully confirm this finding. While some experts appreciated the sketchy design (thanks to the “hand-made” look), others rejected it. This feedback prompted us to add a “sketchiness slider” after the first phase of the study. While the application loads without sketchiness by default, users can now select between no , some , or strong  sketchiness. Fig. 7 shows the system with sketchiness; all other screenshots include no sketchiness. As sketchiness is employed only for locutions in text-based views the risk of it use visually cluttering the workspace is low, but further research is needed to determine which presentation is most effective [4].

The typical size of a corpus annotated for argumentation ranges from 10,000 to 100,000 words. As annotating the transcript of one

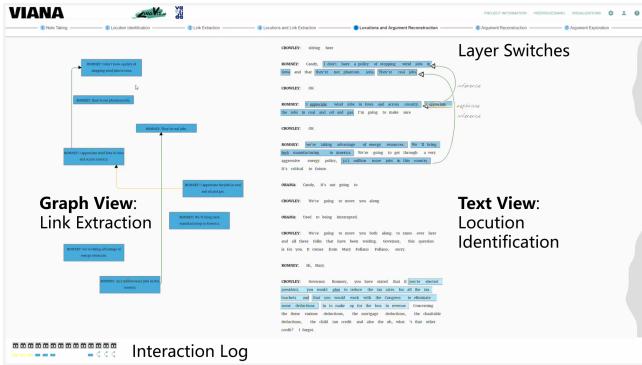


Figure 4: VIANA system overview. The top bar shows the currently active annotation layer. Here, a graph view is shown next to a text view. An interaction log is displayed at the bottom of the screen.

hour of a debate or discussion can take up to fifty hours, annotators usually split this input into manageable chunks, annotate them separately, and carefully merge the intermediate results. The main reason for chunking the input is that it is otherwise difficult to maintain an overview of what has been annotated. VIANA highlights the identified locutions in the input text, enabling experts to relate arguments and their relations to their origin and simplifying the task of keeping an overview. Consequently, annotators can increase the amount of text they load into the application.

Some existing systems rely on manual node placement on the annotation canvas. With increasing sizes of the argumentation graph, more and more time is spent on keeping the canvas organized. While automated layout routines do exist, they are not always as effective as our experts would expect due to the complexity of IAT graphs. With three different argument graph views using automatic layouts and only showing task-relevant information we free users from this strenuous task, enabling them to focus on annotating instead.

All text and graph views mentioned above will be introduced in detail in Sect. 5. In the following section, we introduce the layering and transitions between these views.

#### 4.1 Layered Interface

The previous section has already alluded to the typical sizes of argumentation corpora. While Scheuer et al. claim that scrolling interfaces can cause users to “lose the big picture” [54] they are difficult to avoid in text-based systems. We instead prioritize reducing the amount of information on screen through the introduction of layers. By providing the task-specific layers shown in Fig. 1 users only see the information that is currently relevant to them. By advancing from one task to another on the spectrum, users transition away from text-level views towards a graph-based overview. The different views are intended to enable both close and distant reading. To switch between layers users scroll their mouse wheel while pressing the control key. In order to avoid context switches and make the changes as easy to follow as possible, we smoothly morph all elements on screen. The *Note Taking*, *Locution Identification* and *Link Extraction* views are aligned to minimize positional movement. Barring overlap removal, the respective top-left corners of locution annotations and graph nodes are placed at the same screen position, leaving users with morphing, but stationary rectangles and graph edges. When switching to the *Argument Reconstruction* or *Argument Exploration* views, transitions and targeted scrolling support users in keeping the context. To make changes easier to follow, elements under the mouse remain there after the transition, if possible. This concept is familiar from zooming in maps or image viewers.

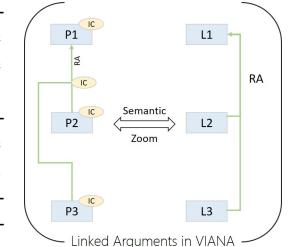
As the layers are organized by task progression and often provide functionality for multiple tasks, frequent back-and-forth switches between layers can be avoided. After the first evaluation phase, we

added two additional layers at the request of some users. They are indicated by dashed lines in Fig. 1 and contain two visualizations side-by-side to enable parallel work on multiple tasks, at the cost of higher information density. As the individual layers remain available, users can select whichever representation is most effective for them in their current context. The system overview in Fig. 4 presents such a combined layer showing both the link extraction and locution identification views at the same time.

We initially decided to introduce layers rather than employing multiple coordinated views or a tabbed or multi-window interface to facilitate relating the resulting argument graph structure to the original text. Scrolling through the layers maps the graph directly into the original text fragments. While linking and brushing in coordinated views could offer similar functionality, it would require at least three views (propositions, locutions, text). Some expert users in our study preferred the layered approach over multiple parallel views as it enabled them to reduce the amount of information to a level they were comfortable with. Heer and Robertson studied animation in statistical data graphics and found that “animated transitions can significantly improve graphical perception.” [27]. We argue that their result “animation is significantly better than static across all conditions” [27] in object tracking tasks is also applicable during layer switches in VIANA. Consequently, we employ a layered approach rather than using tabs or multiple windows.

#### 4.2 Visual Representation of Illocutionary Connectors

Due to the introduction of interface layers, propositions and locutions are not always shown on the screen at the same time. As a result, illocutionary connectors can no longer be rendered like in Fig. 3 (yellow nodes). They are, however, a fundamental part of the underlying Inference Anchoring Theory and need to be represented. We thus map them to the left-hand side of the graph, showing them as badges on both propositions and interpropositional relations between them. While the existence of these connectors is fundamental to the theory, the importance of their particular values is task-dependent and they can often be initialized with sensible default values. VIANA thus initializes connections between locutions and propositions as “Asserting” and those between transitions and inferences as “Arguing”. We provide a setting hiding the illocutionary connectors, allowing users to focus on other tasks and checking the connectors at a different time.



#### 4.3 Automated Suggestions

VIANA highlights keywords that are of specific interest to the annotation based on pre-defined word lists [25]. Connectors like “so”, “if”, “because” or “for” are **bold** and keywords like “appreciate”, “promise” or “complain” that are associated with speech acts are *italicized and underlined*. These highlighting-techniques have been found to work comparatively well in the presence of “distractors” like the boxes around locutions [62].

In addition to highlighting keywords as guidance for manual annotation, we provide proposed fragments of text that should be annotated as locutions. These fragments are discourse units that have been classified as potential locutions by our recommendation system introduced in Sect. 6. Possible interactions like confirming or rejecting suggested locutions follow guidelines for Human-AI interaction [2] and will be introduced together with their influence on future suggestions in Sect. 5.2 and Sect. 6.2, respectively.

Locutions that have been confirmed by users are shown in a dark blue , while those suggested by linguistic rules and predictions based on user-interactions are light blue and teal, respectively. The certainty of suggested annotations is mapped to their

You're absolutely right. You heard what I said about tax plan. The top 5 percent will continue to pay 60 percent, as they do today. I'm not looking to cut taxes for wealthy people. I am looking to cut taxes for middle-income people. And why do I want to bring rates down, and at the same time lower exemptions and deductions, particularly for people at the high end? Because you bring rates down, it makes it easier for small business to keep more of capital and hire people. And for me, this is about jobs. I want to get America's economy going again. Fifty-four percent of America's workers work in businesses that are taxed as individuals. So when you bring those rates down, those small businesses are able to keep more money and hire more people. For me, I look at what's happened in the last

(a) The *Note Taking View* with proposed annotations

You're absolutely right. You heard what I said about tax plan. The top 5 percent will continue to pay 60 percent, as they do today. I'm not looking to cut taxes for wealthy people. I am looking to cut taxes for middle-income people. And why do I want to bring rates down, and at the same time lower exemptions and deductions, particularly for people at the high end? Because you bring rates down, it makes it easier for small business to keep more of capital and hire people. And for me, this is about jobs. I want to get America's economy going again. Fifty-four percent of America's workers work in businesses that are taxed as individuals. So when you bring those rates down, those small businesses are able to keep more money and hire more people. For me, I look at what's happened in the last

(b) The *Locution Identification View* showing the result of an annotation

Figure 5: The two text-based views tailored towards slow analytics and locution identification. In the *Slow Analytics View*, users can read the text and jot down notes. In the *Locution Identification View*, they can mark locutions and introduce relations between their propositions.

opacity. We deliberately chose shades of blue to avoid conflicts with the colors used for relations between arguments. We selected three relatively similar colors to avoid overwhelming users with too much information; an approach that was validated by experts in our user study. An earlier design of the system colored locutions based on the presence of six different types of discourse unit connectors (see Sect. 6.1) as shown on the side. We chose a simplified color scheme based on expert feedback that such information was interesting but not helpful during the annotation process.



## 5 TASK-DRIVEN INTERFACE LAYERS

In the following section, we introduce the layered views that VIANA offers for specific tasks. Several layers offer functionality suited for multiple tasks, and two intermediate layers merge graph- and text views. To transition between layers, users can either select a target layer from a list at the top of their screen or press the control key while using their mouse wheel.

### 5.1 Slow Analytics and Note-Taking

The *Note-Taking* and *Slow Analytics View* represents the “distraction free” mode of VIANA and is presented in Fig. 5a. Depending on user settings, it shows only the raw text or includes the initially proposed locutions. Interviews with experts have revealed different approaches to argument annotation. While one approach starts by immediately annotating locutions, another approach initially scans the text for passages of particular interest. The *Slow Analytics View* offers a note-taking interface—addressing task [T1] presented in the introduction—that allows users to jot down “free-form” notes on fragments of text without being forced to mark them as locutions. This is an advantage over other annotation systems [33] where users employ such tactics to compensate for missing functionality. Once users have gained an overview of the corpus at hand, they progress to the *Text View* for locution identification and relation extraction.



### 5.2 Text Segmentation and Locution Identification

Users transition to the *Locution Identification View* to perform [T2] and annotate locutions. To create a new locution boundary, they select a fragment of the text by clicking and dragging. Once the users let go of the mouse button, both the locution and the corresponding proposition are automatically extracted. The locution is connected to the temporally preceding locution via a transition. However, to avoid cluttering the view, these automatically created transitions are not displayed on the screen and are only contained in the result extracted at the end of the analysis. As an alternative to manual annotation, users can explore the proposed locutions. VIANA displays them in more muted colors and with a lower opacity, as can be seen in Fig. 5a. The colors encode the different origins for these fragments as introduced in Sect. 4.3. An area chart on the right-hand side of the screen summarizes the annotations and can show regions with

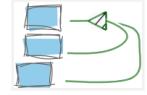
fewer annotations than expected. Those regions might either be of less interest to the analysis or indicate missed locutions.

Every locution displays a toolbar when hovering over it. This toolbar allows confirming ✓ or unconfirming ? a locution, opening the edit 🖍 tooltip to change its type or provide an annotation (as introduced in the slow analytics view), deriving a locution from it 🔍, for example, to resolve reported speech, or deleting it ✘ outright.

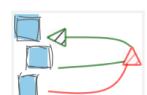
To draw a connection between two locutions, users click and drag from one source locution to a target. Pressing the shift key while dragging will result in a transition while pressing control will result in an inference. A transition is shown as a light grey line — connecting the two locutions directly.

By default, any non-transition edge is drawn as an inference →. Double-clicking on the edge or its label iterates through the available edge types, including conflict → and rephrase →. The associated colors green, red and yellow are well-established in the argumentation community. A toolbar similar to that described for locutions is available for propositional relations as well. It allows to open an edit tooltip 🖍 or delete the edge ✘. Users can change the type of edge in the tooltip using a drop-down menu. Users can also set an *argumentation scheme* [66] in the tooltip. Such a scheme describes the type of the relation more precisely and is displayed instead of the default “inference”, “conflict”, and “rephrase” as an edge-label once selected. Additionally, the tooltip allows users to set the illocutionary connector between the transition and the propositional relation. It will be shown as a badge in the Graph View.

To create a linked argument, users can draw an edge to the arrowhead of an existing propositional relation. This causes the arrowhead to move backward from the end of the edge and receive both incoming edges there. The increased distance between arrowhead and locution emphasizes the merge and clearly distinguishes linked arguments from converging arguments. Converging arguments are achieved by simply drawing multiple edges ending in the same locution. Contrary to linked arguments, the premises of a converging argument can all support or attack the conclusion individually. Consequently, no unique visual mapping showing their connection is necessary or warranted.



To create an undercut, i.e., the (typically) attack or support of an existing propositional relation, a new edge is drawn to the label of an existing edge. Users can link up more than two arguments and undercut or support an existing undercutting relation.



These simple interactions allow users to annotate the text with locutions without having to switch between separate text and graph visualizations. By staying within the same visualization, the context of the original utterances remains available, facilitating the annotation. By overplotting the annotations over the text view, the information density on this level rises progressively during the annotation. Nonetheless, we chose this design to prevent users from having to continuously switch layers or visual representations. Fig. 5b shows

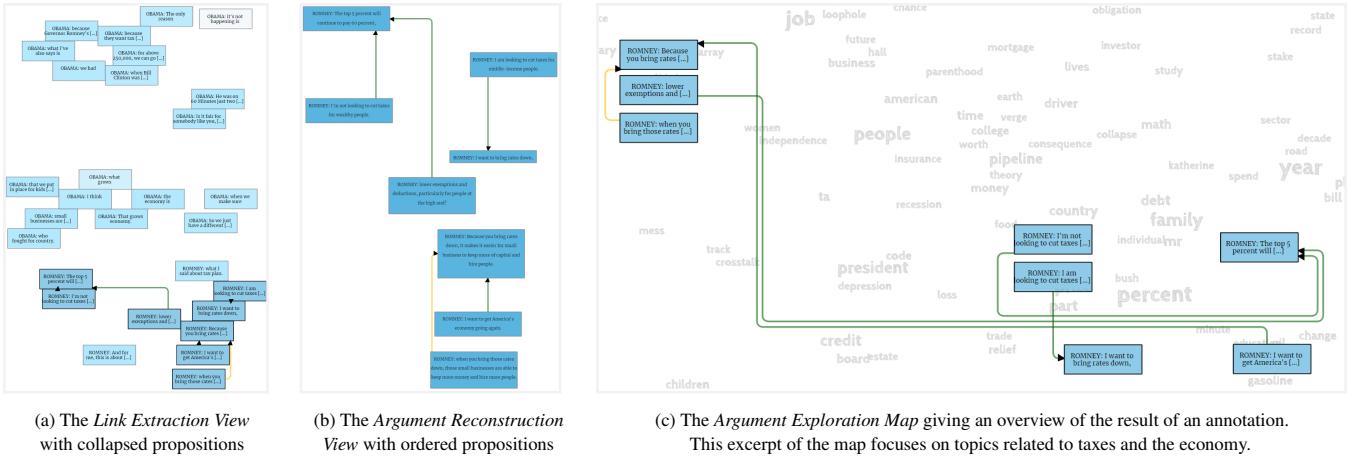


Figure 6: The graph views enable users to explore the extracted propositions. The *Argument Ronstruction View* temporally orders propositions (same data as in Fig. 5) and highlights groups and disconnects in the argument graph. The *Argument Exploration Map* provides a distant-reading overview and shows that the annotated text discussed the impact of tax reductions on people.

typical density of annotation. Fig. 7 shows an utterance by Donald Trump and is denser due to his typical style of speech with many repetitions and short arguments rather than long explanations. Alternative designs would remove either the relation annotation (note taking view) or the text (graph views) to lower the information content displayed on screen and freeing up pixels. As those views are implemented in the system, users are free to choose whichever view fits their workflow best. Such freedom of choice proved important in our user study with experts that have a clear workflow in mind.

### 5.3 Relationship Extraction

While the previously introduced text view also enables the creation of relations, it already contains a lot of information. To enable users to focus on task [T3]—extracting relations between already annotated locutions—VIANA contains the *Link Extraction View* shown in Fig. 6a. The nodes of this graph are propositions (i.e., the left-hand side of an IAT diagram) associated with locutions visible in the text view. To enable smooth transitions between the text and graph view, while avoiding overlap and maintaining readability, this graph shows nodes with shortened text representations at the position of the locution in the text. This positioning allows the outlines of locutions to seamlessly morph into graph nodes, and vice versa. In some cases, minimal position changes are necessary to remove node overlap.

The view provides a quick, compact overview of the current annotations and makes it easy to relate the propositions to their respective locutions as they morph into each other. Hovering over a proposition expands it to fit the entire text. Missing propositional relations can be drawn using the same interactions as in the text view.

To prevent overlap issues we utilize a modified version of Web-CoLa by Dwyer et al. based on their previous work on graph layouting [15]. It produces a gridified layout with no node-edge-overlap and fewer edge crossings than drawing direct links between nodes. The remaining edge crossings are easy to make sense of as they are always intersections of orthogonal lines. The same techniques are applied in the two graph views presented in the following sections.

### 5.4 Argument Reconstruction

While the *Link Extraction View* still contains the propositions of unconfirmed proposed locutions, they are removed in the *Argument Reconstruction View* shown in Fig. 6b. All remaining nodes are expanded to fit the entire proposition. To resolve the resulting overlap issues, the nodes are redistributed along the y-axis. Whenever the nodes of two propositions overlap, the one mentioned in the original text at a later time is shifted downwards until the overlap is resolved.

The result is an automatically arranged timeline graph of propositions that can be read from the top left to bottom right. This temporal alignment identifies connected components of the graph and facilitates detecting breaks and cuts in the annotation.

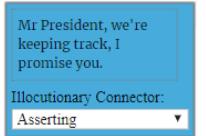
Double-clicking a node enables users to edit a proposition in-place. Changing the text of a proposition is called *reconstruction* ([T4]) and is introduced in Sect. 3. The necessary amount of reconstruction varies depending on the analysis task at hand. Typical changes include rephrasing the locution to form a complete sentence with subject, verb, and objects, lowercasing the first letter, and resolving pronouns. Once users have reconstructed a proposition, the graph nodes show the reconstructed text as their label. The text of the underlying locution is available when hovering over the node.

In addition to reconstructing propositions, users can also change the illocutionary connector when editing a proposition. This connector is initialized with “Asserting”. Depending on the proposition other possible values include “Questioning”, “Restating”, or combinations like “Assertive Questioning”. While the temporally ordered graph with expanded propositions is designed for reconstruction in particular, all three graph views available in VIANA support the task using the same double-click interaction. Hence, any graph view can also be used to introduce new relations between propositions.

The initial version of VIANA showed only one visualization at a given time. After the first study phase we added two layers that combine the *Locution Identification* view on the right and either the *Argument Reconstruction* or *Link Extraction* view on the left, as shown in Fig. 4. This addition is a direct response to requests from some experts wanting to frequently switch between the tasks.

### 5.5 Argument Exploration

The *Argument Exploration Map* provides a distant reading interface enabling exploration ([T5]) by giving an overview of groups of arguments and their respective important keywords. Besides exploration, such a map facilitates the communication of annotation results and enables progress checking during the annotation. It is also useful when continuing the annotation after a short break, as our user study revealed. The graph shown in this view retains all interactive functionality available in the other graph views. The possibility to create new relations in the map view is helpful to create so-called “long-distance relations”. These appear, for example, in political debates where speakers frequently refer back to something that was said some time ago. The topic map reduces the distance between the



propositions if they have similar content, facilitating the creation of a relation, and avoiding scrolling.

To create the map, we first extract all nouns and named entities from the input corpus [17]. Additionally, we gather the top five corpus words each according to word frequency, TF-IDF,  $\hat{G}^2$  and average likelihood ratio. We represent each keyword as a vector using a word embedding tailored to concept words [59] that combines and optimizes the results of word2vec [46] and Glove [48]. We then project these high-dimensional word vectors to 2D using tSNE [63] and fit the created space to the dimensions of the screen. As showing all keywords at once would clutter the screen, we only place those on the map that appear in potential suggested locutions (see Sect. 4.3). After scaling keywords according to their frequency in the text, we obtain a map of the most relevant keywords.

We represent each proposition as a weighted average word embedding vector of nouns and named entities. The initial weights for named entities are twice those for nouns to capture their typical importance. If a proposition contains no keywords that are visible on the topic map, we employ a nearest-neighbor search in the word-embedding space to find appropriate, representative nouns. This step is especially crucial for propositions containing no nouns or named entities. Having obtained a representative aggregated embedding vector for a proposition, we can determine its position in the projection. While we prevent node-overlap in the map by nudging nodes apart, we cannot prevent all node-keyword overlap. As a consequence, we offer a toggle switch that swaps the z-order of keywords and nodes to reveal keywords hidden behind nodes.

## 5.6 Interaction Log

The interaction timeline at the bottom of the screen summarizes the annotation session and is visible from every layer. It shows information about identified and deleted locutions . Employing the same visual metaphors, it also informs about added , re-typed and deleted relations between both locutions and propositions. All colors correspond to the colors of the affected entities at the time of each interaction. The timeline reveals different annotation patterns that also became apparent in our expert user study: some users identify locutions and directly connect them with links wherever possible, while others focus on extracting all locutions first and create links in a second step. The confirmation and rejection of locutions as well as annotation changes, including those to illocutionary connectors and argumentation schemes are displayed as well.

The timeline does not show any contextual information for the recorded annotation changes. According to the experts we consulted, this is not necessary as they are aware of the changes they performed during the last few minutes. However, they did express the wish to keep modifying the respective elements directly from the timeline, for example, to adjust the previous interaction based on a new insight. The interaction tracking serves as a data collection method for further improvements to VIANA. While the current version uses “locution interactions” to improve suggestions, future versions of the system will use the data to provide undo and redo actions. Furthermore, the timeline is a first step towards providing analytic provenance.

## 6 INTERACTION-DRIVEN ANNOTATION SUGGESTIONS

To guide and support users in their analysis process, VIANA highlights important keywords and recommends text fragments that should be annotated as locutions. While linguistic rules provide a good basis for initial suggestions, there is great potential for efficient human-machine collaboration. As there is generally not enough training data for good-quality classifiers or fully-automated annotation systems, VIANA utilizes the knowledge encoded in a language model and refines it over time to train a measure of argument similarity from user interactions.

We employ a BERT [12] model that was obtained by fine-tuning the “BERT-Base” checkpoint for evidence- and claim-detection. We gathered the training samples from several IBM Project Debater training datasets [1, 52, 55] and chose the BERT base model because of its state-of-the-art performance. However, the recommendation component in VIANA is not specific to BERT and could also be used with embeddings from any other language model.

The system first enumerates so-called “elementary discourse units” [25]—(sub)sentences delimited by punctuation and clausal connectors—as *fragments*. Each fragment is represented as a tuple  $(embedding, label, weight, points, state)$ . Besides the embedding  $e$  the tuple contains a label  $l \in \{-1, 0, 1\}$  that indicates whether a fragment should be an annotation ( $l = 1$ ), not be an annotation ( $l = -1$ ) or is still undecided ( $l = 0$ ). The weight  $w$  is used in similarity computations between two fragments. It is initialized to  $w = 1$  and updated through user interactions as described in Sect. 6.2. The points are initialized to  $p = 5$  and decay over time if users ignore suggested fragments but interact in their vicinity. The state  $c$  of a fragment can be either CREATED, LINGUISTICS or CONFIRMED.

VIANA only recommends locutions for annotation, not relations between them or their extracted propositions. While this is a field that we will explore in future work, discussions with annotation experts showed that they are more reserved with respect to proposed relations than proposed locutions. This skepticism stems from the fact that annotating relations is a significantly more complex task, and experts prefer to do a good job manually rather than having to correct imperfect suggestions. Therefore, we do not include proposed relations yet.

## 6.1 Initializing Suggestions

To avoid a cold-start of the recommendation system, we initialize it with the output of a linguistic discourse-unit annotation pipeline [25] and refine it throughout the annotation process. The pipeline identifies discourse units that have a connection of type *conclusion*, *reason*, *condition* or *consequence* to the surrounding discourse units. We add all discourse units that contain a speech act of type *agreement* or *disagreement* and set the labels for the suggestions to 1 and their state to LINGUISTICS. All other discourse units are labeled 0 and remain in state CREATED.

To suggest fragments to users, we compute the score  $s$  for each fragment  $f_i$  as the average of the weighted cosine similarity:

$$s_i = \frac{1}{j} \cdot \sum_{j \in C} \cos(e_i, e_j) \cdot w_j \cdot l_j \cdot p_i$$

where  $C$  is the set of fragments  $f$  such that  $C = \{f \mid f_c = \text{CONFIRMED} \mid |f_c = \text{LINGUISTICS}\}$  and  $\cos(a, b)$  the cosine similarity between  $a$  and  $b$ . Already (partly) decayed points  $p_i$  and a higher similarity with fragments  $f_j$  with a negative label  $l_j$  lead to a lower score, and hence a lower probability of  $f_i$  being suggested as an annotation.

We chose cosine similarity over the dotproduct, and normalization factor  $\frac{1}{j}$  over  $(\sum_{j \in C} w_j)^{-1}$  as this combination led to the best separation between confirmed and rejected suggestions in our tests.

After sorting all fragments with  $c = \text{CREATED}$  according to their score we return the  $n$  suggestions with the highest score. Any fragments with  $c = \text{LINGUISTICS}$  are shown as suggestions (if they have not been manually deleted), independent of their score. While the correct number of  $n$  depends on the datasets, our study participants noted they wanted rather more than fewer suggestions.

## 6.2 Promotion and Decay of Suggestions

To refine the suggestions over time, we update the weights and labels of fragments through user interaction. When users confirm or disconfirm (i.e., mark as a draft) a locution, we triple the weight of the associated fragment or divide it by three, respectively. Deleting a locution leads to the weight of the associated fragment being doubled, while the label is changed from 1 to  $-1$ . Manually added

locutions are initialized with a weight of 4. These weights and updates ensure that items that have been interacted with take a more important role in the similarity calculation. Compared to confirmed locutions we keep the weights for deleted locutions lower to ensure that the system keeps proposing fragments that should be annotated, rather than those that should not be. The concrete values of the weights were identified through initial experimentation. Their general distribution (lowest values for negative feedback, highest values for manual intervention) follows our previous work on model optimization through progressive learning [18].

To avoid cluttering the screen with suggested annotations, we introduce a *viewport-dependent suggestion decay* function. While there are simple ways to learn from direct user-interaction as described above, the number of interactions in a system is limited. VIANA, thus, also learns from the items that users do not interact with. Recall the points  $p$  associated to every fragment. Whenever users interact with an item, suggestions that are close on screen but are not interacted with, lose some points. This process captures that the suggestion was not relevant to the users, and they rather performed a different action. We calculate the point loss of fragment  $f_i$  after an interaction with fragment  $f_j$  as  $pl_i = \log_{32} D - \log_{32} d(f_i, f_j)$  where  $D$  is the maximum (absolute) decay distance and  $d(a, b)$  the distance in the text as number of words between fragments  $a$  and  $b$ . The maximum decay distance  $D$  is dependent on the amount of text visible on screen and ensures that only those suggestions that are visible to users and likely to be “interaction alternatives” lose points. VIANA currently sets  $D$  to 200. As a result, each fragment can lose, in theory, at most 1.5 points after each interaction. In practice, the maximum loss is closer to 1 due to the typical length of a locution. We update the points of visible suggestions after each interaction, with the exception of confirming suggestions.

Once a fragment lost all points, we set its weight to 2, its state to CONFIRMED, and its label to  $-1$ , mimicking a user manually deleting the suggestion. Fragments that lose all points are taken into consideration as negative samples when generating new suggestions. This novel approach for viewport-dependent suggestion decay differs from previous work on suggestion decay in recommender systems that is typically based on temporal evolution or ignores on-screen context. Temporal decay is not suitable for argumentation annotation as there are no changes to annotation guidelines during an individual annotation. Our approach penalizes individual items in addition to updating the content-based similarity function. It incorporates the on-screen distance between ignored suggestions and interactions to inform the speed of decay. This takes into account that users are likely to be much more aware of the content in the direct vicinity of their interaction, especially in text-based systems.

## 7 EVALUATION

To validate the effectiveness of our approach, we conducted an expert user study with five participants over a period of three months. We present its results after introducing two independent use-cases that demonstrate the usefulness and practical applicability of VIANA.

Expert E2 (who will be introduced in the following section) validated the choice to forgo a quantitative evaluation and stated that inter-annotator agreement studies for argumentation often fail because of the multi-stage process of annotation. Once annotators disagree in the identified locution structure, their propositions, relations or argumentation schemes are automatically not in accordance.

Furthermore, the time needed for annotation is not necessarily a useful metric as experts already spend many hours annotating and prefer exact over fast results. Much time is spent on reasoning about the underlying argument structure during annotation. Consequently, computing speed-up factors between two subsequent annotation runs in different annotation systems is meaningless. As the argumentative structure will have been identified during the first annotation, the second one will always be easier. Hence, we present qualitative

Figure 7: One utterance by D. Trump from the first Presidential Debate between him and H. Clinton in 2016. His constant rephrasing is made obvious by the sheer amount of yellow edges on screen.

feedback from five study participants. In particular, we highlight their feedback with respect to the design and usability of the system, as well as the usefulness and quality of the suggestions.

### 7.1 Use Cases

Before providing qualitative feedback from our expert user study, we present two use-cases that highlight the general usability of VIANA.

#### 7.1.1 Argumentation Annotation

When loading VIANA to annotate the second presidential debate between Obama and Romney in 2012, the user first sees the close reading view. She begins to read until she finds some utterances on taxes that she decides to annotate. As soon as she begins annotating ([T2]), she is presented with automated suggestions like those shown in Fig. 5a. As she finds the annotation suggestion for “The top 5 percent will continue to pay 60 percent” helpful and decides to accept it, she immediately sees updated suggestions. After annotating some locutions, she begins to introduce relations between them ([T3]) and later switches to the *Argument Reconstruction View*. There, she reconstructs propositions ([T4]) to fix grammatical errors, remove trailing punctuation and turn questions into statements. Next, she investigates the argumentative structure of the annotated segment. The yellow rephrase relation highlights Romney’s plan to lower taxes. Using her scroll wheel to progress to the *Argument Exploration Map*, the user finds that the debate also seems to discuss college education in the context of taxes and the economy ([T5]). She decides to investigate this topic more closely and figure out how additional arguments relate to those already identified. Switching back to the *Locution Identification View* she soon finds an exchange between Obama and an audience member on tax reduction through education credits that she annotates next.

#### 7.1.2 Argumentation Behaviour

Moving on to the first presidential debate between Hillary Clinton and Donald Trump, we can use the annotation of argumentation to observe different argument behavior strategies. Fig. 7 shows a single utterance by Donald Trump that a user annotated in VIANA. Trump is attacking Hillary Clinton by claiming that her experience as Secretary of State is “*bad experience*.” He then keeps repeating and rephrasing this claim through the entire utterance. While he occasionally mentions other facts or makes new claims, these are not connected to other propositions with inference or conflict relations. Instead, he repeatedly restates his opinion on her qualifications.

In such a scenario, displaying the annotation directly in the text view yields valuable insights. It is easy to see that the utterance is intra-connected with rephrases, while still containing locutions without any relations. The text view clearly communicates this finding as the close packing of the locutions shows the immediate context of each rephrase. A graph view containing only locutions or propositions would likely fail to transmit the same image due to the lack of textual context. As a consequence, it would not become immediately obvious that all annotations originated from a single utterance, or whether long breaks separated the annotated fragments.

Similar phenomena can be observed when discussion participants make unsubstantiated and unrelated claims or contradict themselves within the context of a single utterance or a short period of time.

## 7.2 Expert User Study

In addition to the use-cases presented above, we evaluate *VIANA* in a two-stage Expert User Study carried out in pair analytics sessions [35]. In each session, one domain expert and one visual analytics expert (one of the authors) were present. The two stages of the study were performed three months apart. The system evaluated in the second stage incorporates expert feedback from the first study.

### 7.2.1 Methodology

We divided the 90-minute long study sessions into three parts. In the first 20 minutes, we presented the system to the expert and explained the functionality. We also elicited initial feedback on the design-choices and usefulness of *VIANA* through a semi-structured interview. In the following 40 minutes, we gave the expert control over the system interface and let them explore the dataset. We supported the expert with clarifications on the functionality of the user interface and the controls whenever they had questions. Occasionally we also proposed to progress to a different view to ensure that the expert got a holistic impression of the system. Each expert was encouraged to think aloud and explain the rationale for their actions. After the exploration period, we transitioned into another semi-structured interview of 30 minutes. Here we asked the expert for detailed feedback based on their experience with the system to receive a general assessment. We focussed on the design, the usefulness of the tool to the expert, the quality of suggested annotations, and potential missing features. We recorded both audio and video from the screen during all study sessions.

**Participants –** **E1** holds a Ph.D. in computational linguistics and currently works as a postdoctoral researcher. As argumentation has a crucial role in her research, she currently spends multiple hours a day annotating argumentation data. She also teaches courses about argumentation and the underlying theory. **E2** just completed his Ph.D. at the intersection of computer science and computational linguistics, working on argumentation and ethos mining. He estimates to have worked on manual argumentation annotation for over one full year in the last 3.5 years of his Ph.D. During the design phase, E2 provided his domain knowledge about less common IAT annotations that he felt should be possible in *VIANA*. As a consequence, he had seen, but never used, the system before the study. He did not contribute to the visual design, the interaction design, or any of the analysis layers in particular.

Participants **S1**, **S2**, and **S3** are masters students in Speech and Language Processing, have received special training in argumentation annotation over six months and work as student assistants in argumentation annotation now. E1 and E2 participated in the first, and S1, S2, and S3 in the second phase of the study. None of the experts are authors of this paper.

**Dataset –** As datasets, we chose transcripts of presidential debates as all experts had experience with annotating political debates. In the first study phase, we used the second 2012 debate between Barack Obama and Mitt Romney because E2 had previously annotated all three of the more recent debates between Trump and Clinton. In the second phase of the study, we used the first 2016 debate between Trump and Clinton. None of the experts had annotated the respective data used in the study before. All participants were presented with 40 utterances from the end of the debates. This section of the text has been selected to skip the non-argumentative introduction phase of the debate and fit the text-length to a typical annotation session that our experts are used to.

### 7.2.2 Results and Feedback

In this section, we report the feedback from the three phases of our study sessions. We summarize the comments of the participants, providing a selection of the most insightful feedback.

**Initial Feedback –** Both E1 and E2 highlighted the importance of manually annotated argumentation for their research. None of

the five experts had previously worked with visual analytics systems for argumentation annotation. However, they were excited about the idea of working with the proposed locutions, with E2 stating “*I think this would speed up the whole process [of annotating] quite significantly.*” S1 agreed and articulated “*I wish I just had to check if the locutions were already extracted correctly.*” E1 highlighted the importance of human-in-the-loop analysis. She liked the idea of proposed locutions “*as long as I can change them.*” This response mirrors the generally reserved attitude of the experts towards fully automated approaches. At the same time, all experts expressed that they were aware that suggested annotations might bias the result towards the suggestion, especially in hard-to-decide situations.

While she thought the extracted tasks were relevant and captured her actual work, E1 was skeptical of the individual analysis layers we introduce for each task. As her recent work has often focussed on illocutionary connections, she stated to be “*suspicious about the illocutionary structure.*” S2 had similar skepticisms at first. However, she changed her opinion after being introduced to the system and mentioned she thought “*it is more manageable.*” S3 agreed that it is “*good to see the different layers. Sometimes it makes things easier to understand.*” One of E2’s biggest complaints about the system he currently uses is that “*if you do not constantly move nodes around to make it look good you lose track of what is going on*” and that “*significant time is spent making the graph readable.*” Consequently, he liked the idea of automated layouts and separated layers introducing a more rigid visual structure to the graph. Later system use revealed two annotation patterns. While some users directly interconnect extracted locutions, others first extract some locutions, before transitioning through the layers to link and reconstruct them. They then transitioned back to the text view, effectively annotating data in “mini-batches”.

**Design and Usability –** When presented with the system for the first time E1 remarked that the sketchy-rendering of nodes and relations might be confusing and could suggest “*that you are not actually sure what you are doing.*” She argued that she was performing a “*precise analysis*” that did not warrant a sketchy representation. In her expectation, only proposed locutions should have been sketchy. This sketchiness should then have been removed once users confirmed the entities or never be introduced for those created manually. E2 was more fond of our design choice and described the visual design as “*modern.*” He also felt that the sketchy rendering made him more inclined to keep editing the annotation as opposed to existing systems where he feels like no changes can be made anymore. As a reaction to these opinions, we made the sketchiness configurable for the second phase of the study. Here, S3 chose the sketchy version because it felt “*just like working on a piece of paper*”, while S2 dismissed the sketchy option saying “*it does not feel as official.*”

Both S1 and S2 were concerned with the superposition of the extracted locutions and the text, unanimously calling it “*very dense.*” They later stated that they preferred the offered graph views to create new links between arguments. After using the system, S2 felt that the design was “*not cramped*” and that the “*highlighted parts seem clearly separated.*” Additionally, she now thought the superposition made it easier to add, delete, or refine locutions. In the current version of the system, users have to remove irrelevant suggestions manually or wait for them to decay over time. S3 wanted the system to automatically remove suggested annotations as soon as she manually annotated an overlapping locution, a request that we plan to include in future work.

**General Assessment –** The experts unanimously praised the system for its proposed locutions and the potential they bring for reducing the overall time needed for annotation. They did also have ideas for small improvements, like resizing an already identified locution to add or remove individual words from the beginning or end. S1 noted that having suggestions “*makes things a lot easier.*” However, she also noted that a careful trade-off had to be made between

showing too many and too little suggestions and matching the users' expected density. In the subsequent study, S2 reached three consecutive sentences without suggestions and exclaimed "*it makes me wonder whether I am over-annotating.*" As she continued annotating suggestions appeared in the previously empty area, leading her to conclude that the system was learning from her interaction.

After her annotation session, S2 began to reason about the quality of suggestions. She liked "*that [VIANA] has two 'tracks' of suggestions*", referring to the different colors assigned to suggestions from the linguistic pipeline and the learned user interactions and said she was "*trying to figure out which ones [she] trust[s] more.*" She concluded that she found the linguistic suggestions more reliable in the beginning, but would become more reliant on those learned from her interactions over time. E1, S1 and S2 expressed that they did not feel that the suggestions had biased the annotation result, with S1 saying "*you still need to think about the suggestions.*"

While none of the experts found the *Argument Exploration* view essential during the annotation of a section of text, E1 stated she found it useful to communicate the results to colleagues. E2 liked it in particular as a means to introduce long-distance relations between nodes that would be far apart in the other views. He also imagined using it whenever coming back to the annotation after a break to get back into the context quickly. S3 found the map "really useful" when annotating longer texts and envisioned that you could "use it to check on yourself" and observe your progress.

Summarizing her experience with the system, computational linguist expert E1 said "*This is really nice, and it will help a lot.*" One important factor for her was the fact the VIANA enables her to load and annotate larger amounts of data at once: "*I definitely like that you can put in a lot of text.*" This saves time that is otherwise spent on combining the results of the annotation of multiple chunks. However, she was not in favor of splitting the locutions and propositions onto two different analysis layers, preferring a side-by-side view instead of having to switch layers during the annotation process, as this would be "*distracting.*" She stated "*we want locutions and propositions next to each other*" and suggested introducing a fifth layer in the middle of our current set of views. We have since added such a combined view after the first phase of the study. E1 also expressed that she would still use the other views before and after the annotation process, naming the note-taking interface ("*it would be really great to have [VIANA], also because of the comment functionality*") and the topic map to communicate the final results.

After the study had ended, E2 remained seated in front of the system and kept transitioning between the layers, saying "*this is so cool!*", validating our approach of bringing visual analytics to argumentation annotation. While the expert users have strong mental models reinforced through the long-time use of existing systems, they are open to new developments. Despite the learning period of new systems, it is promising that users found it engaging in the little time they had with the system.

### 7.3 Discussion and Lessons Learned

The expert user study highlighted the demand for a flexible tool that is capable of adapting to the users' needs and expectations. While they are interested in new systems and eager to try them out, they do have very specific layouts and functionalities in mind. It is only through personalization of the interface and exactly understanding their work processes that we can provide them with an efficient system [29]. The current version of VIANA makes the *sketchiness* of the tool configurable, giving users the option to keep it active at various degrees of intensity or disable it outright. As the annotation process is subjective and highly time-consuming, we need to ensure that we cater to the different mental models of users to create a good user experience for them.

Users already liked the relatively high degree of automation that we provide. Despite being generally reserved concerning fully-

automated argumentation mining tools, the experts were more at ease once they knew that they would still be able to manually "overwrite" the system later. The general wish for more automation became apparent when S1 asked "*do I need to do this manually?*" when changing an illocutionary connector. She was already comfortable with the interaction model of suggested locutions and would have preferred to confirm a suggested change here as well. Summarizing all study sessions, the list of requested automation steps includes the resolution of reported speech and pronouns, the lowercasing of propositions, selection of argumentation schemes, or "*hiding*" non-argumentative areas of text.

As the pre-usage interviews revealed, all experts were aware of potential bias introduced by suggestions. None of them did, however, feel that their final annotation was biased. Consequently, future research is needed to determine whether, and to what degree, users are exposed to various kinds of bias. The study showed not only the effectiveness of our learning approach but also trust-building processes facilitated by simple interaction principles. Before using the system, most users were unsure about the quality of suggestions. After performing a few interactions and realizing that they had full control, they started to build trust and looked forward to the next suggestions, showcasing a successful human-machine-collaboration.

**Limitations –** The design of any system is often a trade-off between expressiveness, complexity, and ease of use. Using layers and semantic transitions, we aim to reduce the complexity of the system and make it intuitive to use. Our evaluation shows that some users have existing workflows that are not well-suited to such a layered approach, while others preferred the reduced complexity. As VIANA provides various layers and combinations of views, users can customize their workspace to their tasks and needs. The current implementation of VIANA is tailored towards texts of up to 10,000 words and thus suitable for typical text lengths in argumentation annotation, according to experts. The argument exploration view becomes crowded for longer texts and would require additional navigation interactions to accommodate the space requirements. Furthermore, annotation is still a manual process and does consequently not scale to very large amounts of data. With increasing text length and, more importantly, increasing annotation density, views like Fig. 7 become more typical. An even denser annotation is theoretically possible but is not likely on real data in an expert-user system. In future work, we will investigate grouping of arguments to enable focusing on particular areas of the data. Showing and hiding such groups can enable scaling to larger datasets. Further, we plan to utilize the design element of sketchiness to encode (non-continuous) information.

## 8 CONCLUSION

We have presented VIANA, a web-based, integrated approach that enables both the annotation of argumentation, as well as the exploration of the results. It provides stacked, task-specific interface layers that we connect with smooth semantic transitions. VIANA automatically suggests fragments of text for annotation and lays the foundation for an extensible platform that will be developed towards semi-automated argumentation annotation and argument mining. The suggestions are refined over time by learning from both the presence and absence of user interaction, introducing a novel approach to suggestion decay. The web-based architecture of VIANA makes it easy to distribute it to new users and opens possibilities for further extensions towards a remote-collaborative tool with which multiple users can annotate individual segments of text at the same time. This is especially interesting in fast-paced environments like the live annotation of radio- or TV-shows. VIANA will be made available as part of lingvis.io [16] under <http://viana.lingvis.io>.

## ACKNOWLEDGMENTS

This work has been funded by the DFG with Grants 350399414 and 376714276 (VALIDA/SPP-1999 RATIO).

## REFERENCES

- [1] E. Aharoni, A. Polnarov, T. Lavee, D. Hershovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics. In *Proc. Workshop on Argumentation Mining at ACL*, pp. 64–68. Association for Computational Linguistics, Stroudsburg, PA, USA, 2014. doi: 10.3115/v1/W14-2109
- [2] S. Amershi, D. Weld, M. Vorvoreanu, A. Journey, B. Nushi, P. Collison, J. Suh, S. Iqbal, P. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz. Guidelines for Human-AI Interaction. In *Proc. Conf. Human Factors in Computing Systems*, pp. 1–19, 2019. doi: 10.1145/3290605.3300233
- [3] Aristoteles and G. A. Kennedy. *On rhetoric: A theory of civic discourse*. Oxford University Press, New York, 1991.
- [4] M. Behrisch, M. Blumenschein, N. W. Kim, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, T. Schreck, D. Weiskopf, and D. A. Keim. Quality metrics for information visualization. *Computer Graphics Forum*, 37(3):625–662, 2018. doi: 10.1111/cgf.13446
- [5] R. Bembenik and P. Andruszkiewicz. Towards Automatic Argument Extraction and Visualization in a Deliberative Model of Online Consultations for Local Governments. In *Advances in Databases and Information Systems*, pp. 74–86. Springer International Publishing, 2016. doi: 10.1007/978-3-319-44039-2\_6
- [6] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, and G. Gorrell. GATE Teamware: A web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029, 2013. doi: 10.1007/s10579-013-9215-6
- [7] K. Budzynska, M. Janier, J. Kang, C. Reed, P. Saint-Dizier, M. Stede, and O. Yaskorska. Towards Argument Mining from Dialogue. In *Frontiers in Artificial Intelligence and Applications*, pp. 185–196, 2014. doi: 10.3233/FKI-144367-185
- [8] K. Budzynska and C. Reed. Whence Inference? Technical report, University of Dundee, 2011.
- [9] E. Cabrio, S. Tonelli, and S. Villata. From Discourse Analysis to Argumentation Schemes and Back: Relations and Differences. In *Workshop on Computational Logic in Multi-Agent Systems at LPNMR*, pp. 1–17, 2013.
- [10] W.-T. Chen and W. Styler. Anafora: A Web-based General Purpose Annotation Tool. In *Proc. NAACL HLT Demo. Session*, pp. 14–19. ACL, Atlanta, Georgia, 2013.
- [11] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, pp. 1–16, Oct 2018.
- [13] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. LeadLine: Interactive Visual Analysis of Text Data through Event Identification and Exploration. In *Proc. Conf. Visual Analytics Science and Technology*, pp. 93–102, 2012. doi: 10.1109/VAST.2012.6400485
- [14] J. Douglas and S. Wells. Monkeypuzzle - Towards Next Generation, Free and Open-Source, Argument Analysis Tools. In *Proc. Workshop on Computational Models of Natural Argument at ICAIL*, pp. 50–53. London, 2017.
- [15] T. Dwyer, Y. Koren, and K. Marriott. IPSep-CoLa: An Incremental Procedure for Separation Constraint Layout of Graphs. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):821–828, 9 2006. doi: 10.1109/TVCG.2006.156
- [16] M. El-Assady, W. Jentner, F. Sperrle, R. Sevastjanova, A. Hautli-Janisz, M. Butt, and D. Keim. lingvis.io - A Linguistic Visual Analytics Framework. In *Proc. of Association for Computational Linguistics, ACL System Demonstrations*. ACL, 2019.
- [17] M. El-Assady, R. Kehlbeck, C. Collins, D. Keim, and O. Deussen. Semantic concept spaces: Guided topic model refinement using word-embedding projections. *IEEE Transactions on Visualization and Computer Graphics*, 2019.
- [18] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive Learning of Topic Modeling Parameters: A Visual Analytics Framework. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):382–391, 2018. doi: 10.1109/TVCG.2017.2745080
- [19] M. El-Assady, F. Sperrle, O. Deussen, D. Keim, and C. Collins. Visual Analytics for Topic Model Optimization based on User-Steerable Speculative Execution. *IEEE Trans. on Visualization and Computer Graphics*, 25(1):374–384, 1 2019. doi: 10.1109/TVCG.2018.2864769
- [20] A. Endert, P. Fiaux, and C. North. Semantic Interaction for Visual Text Analytics. In *Proc. Conf. Human Factors in Computing Systems*, CHI ’12, pp. 473–482. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2207741
- [21] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The State of the Art in Integrating Machine Learning into Visual Analytics. *Computer Graphics Forum*, 36(8):458–486, 12 2017. doi: 10.1111/cgf.13092
- [22] J. A. Fails and D. R. Olsen. Interactive machine learning. In *Proc. Int. Conf. Intelligent User Interfaces*, p. 39. ACM Press, New York, New York, USA, 2003. doi: 10.1145/604045.604056
- [23] J. Fulda, M. Brehmer, and T. Munzner. Timelinecurator: Interactive authoring of visual timelines from unstructured text. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):300–309, 2015.
- [24] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 3 2014. doi: 10.1145/2523813
- [25] A. Hautli and M. El-Assady. Rhetorical Strategies in German Argumentative Dialogs. *Argument & Computation*, (2):153–174, 8 2017. doi: 10.3233/AAC-170022
- [26] C. He, D. Parra, and K. Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56:9–27, 9 2016. doi: 10.1016/j.eswa.2016.02.013
- [27] J. Heer and G. Robertson. Animated Transitions in Statistical Data Graphics. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1240–1247, 11 2007. doi: 10.1109/TVCG.2007.70539
- [28] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. Int. ACM SIGIR Conf. on Research and development in information retrieval*, pp. 230–237. ACM Press, New York, New York, USA, 8 1999. doi: 10.1145/312624.312682
- [29] U. Hinrichs, M. El-Assady, A. Bradley, S. Forlini, and C. Collins. Risk the Drift! Stretching Disciplinary Boundaries through Critical Collaborations between the Humanities and Visualization. In *Workshop on Visualization for the Digital Humanities at VIS*, pp. 1–5, 2017.
- [30] U. Hinrichs, H. Schmidt, and S. Carpendale. EMDialog: Bringing information visualization into the museum. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1181–1188, 2008. doi: 10.1109/TVCG.2008.127
- [31] M. Isik and H. Dag. A recommender model based on trust value and time decay: Improve the quality of product rating score in E-commerce platforms. In *2017 IEEE Int. Conf. on Big Data*, pp. 1946–1955. IEEE, 12 2017. doi: 10.1109/BigData.2017.8258140
- [32] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. In R. Borgo, F. Ganovelli, and I. Viola, eds., *Eurographics Conf. Visualization - STARs*. The Eurographics Association, 2015. doi: 10.2312/eurovisstar.20151113
- [33] M. Janier, J. Lawrence, and C. Reed. OVA+: An argument analysis interface. In *Proc. Computational Models of Argument*, vol. 266, pp. 463–464, 2014.
- [34] D. Jannach, L. Lerche, and M. Zanker. *Social Information Access: Systems and Technologies*, chap. Recommending Based on Implicit Feedback, pp. 510–569. Springer International Publishing, Cham, 2018.
- [35] L. T. Kaastra and B. Fisher. Field Experiment Methodology for Pair Analytics. In *Proc. Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization at VIS*, BELIV ’14, pp. 152–159. ACM, New York, NY, USA, 2014. doi: 10.1145/2669557.2669572
- [36] S. Koch, M. John, M. Wörner, A. Müller, and T. Ertl. VarifocalReader - In-depth visual analysis of large text documents. *IEEE Trans. on Visualization and Computer Graphics*, 2014. doi: 10.1109/TVCG.2014.2346677
- [37] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. Int.*

- Conf. on Knowledge Discovery and Data Mining*, p. 447. ACM Press, New York, New York, USA, 2009. doi: 10.1145/1557019.1557072
- [38] J. R. Landis and G. G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977. doi: 10.2307/2529310
- [39] L. Lerche and D. Jannach. Using graded implicit feedback for bayesian personalized ranking. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys ’14, pp. 353–356. ACM, New York, NY, USA, 2014. doi: 10.1145/2645710.2645759
- [40] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 1 2003. doi: 10.1109/MIC.2003.1167344
- [41] M. Lippi and P. Torroni. Argumentation Mining: State of the Art and Emerging Trends. *ACM Trans. Internet Technology*, 16(2):10:110:25, 3 2016. doi: 10.1145/2850417
- [42] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, and D. Keim. Bridging text visualization and mining: A task-driven survey. *IEEE Trans. on Visualization and Computer Graphics*, 2018.
- [43] Y. Liu, Z. Xu, B. Shi, and B. Zhang. Time-Based K-nearest Neighbor Collaborative Filtering. In *IEEE Int. Conf. on Computer and Information Technology*, pp. 1061–1065. IEEE, 10 2012. doi: 10.1109/CIT.2012.217
- [44] P. Lops, M. de Gemmis, and G. Semeraro. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, pp. 73–105. Springer US, Boston, MA, 2011.
- [45] Y. Lu, H. Wang, S. Landis, and R. Maciejewski. A visual analytics framework for identifying topic drivers in media events. *IEEE Trans. on Visualization and Computer Graphics*, 24(9):2501–2515, 2017.
- [46] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *Proc. Int. Conf. Learning Representations*, pp. 1–12, 2013.
- [47] D. Parra, A. Karatzoglou, I. Yavuz, and X. Amatriain. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *Proc. of CARS*, pp. 1–5, 2011.
- [48] Pennington Jeffrey., R. Socher, and and Manning Christopher. Glove: Global Vectors for Word Representation. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 1532–1543. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1162
- [49] C. Reed and T. Norman. *Argumentation Machines: New Frontiers in Argument and Computation*, vol. 9. Springer Science & Business Media, 1 ed., 2003. doi: 10.1007/978-94-017-0431-1
- [50] C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *Int. J. Artificial Intelligence Tools*, 13(4):961–979, 2004. doi: 10.1142/S0218213004001922
- [51] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, UAI ’09, pp. 452–461. AUAI Press, Arlington, Virginia, United States, 2009.
- [52] R. Rinott, L. Dankin, C. Alzate Perez, M. M. Khapra, E. Aharoni, and N. Slonim. Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 440–450. Association for Computational Linguistics, Stroudsburg, PA, USA, 2015. doi: 10.18653/v1/D15-1050
- [53] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. Int. Conf. on World Wide Web*, pp. 285–295. ACM Press, New York, New York, USA, 2001. doi: 10.1145/371920.372071
- [54] O. Scheuer, F. Loll, N. Pinkwart, and B. M. McLaren. Computer-supported argumentation: A review of the state of the art. *Int. J. Computer-supported Collaborative Learning*, 5(1):43–102, 2010. doi: 10.1007/s11412-009-9080-x
- [55] E. Shnarch, C. Alzate, L. Dankin, M. Gleize, Y. Hou, L. Choshen, R. Aharonov, and N. Slonim. Will it Blend? Blending Weak and Strong Labeled Data in a Neural Network for Argumentation Mining. In *Proc. An. Meeting of the Association for Computational Linguistics (Short Papers)*, pp. 599–605. Melbourne, Australia, 2018.
- [56] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing. Machine Teaching: A New Paradigm for Building Machine Learning Systems. 7 2017.
- [57] M. Skeppstedt, C. Paradis, and A. Kerren. PAL, a tool for Pre-annotation and Active Learning. *J. for Language Technology and Computational Linguistics*, 31(1):81–100, 2016.
- [58] N. Slonim. Project Debater. In *Proc. Computational Models of Argument*, p. 4, 2018. doi: 10.3233/978-1-61499-906-5-4
- [59] R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Conf. Artificial Intelligence*, pp. 4444–4451, 2017.
- [60] C. Stab, C. Kirchner, J. Eckle-Kohler, and I. Gurevych. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *CEUR Workshop Proceedings*, 2014.
- [61] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proc. Demo. Conf. European Chapter of the Association for Computational Linguistics*, EACL ’12, pp. 102–107. Association for Computational Linguistics, Stroudsburg, PA, USA, 2012.
- [62] H. Strobelt, D. Oelke, B. C. Kwon, T. Schreck, and H. Pfister. Guidelines for Effective Usage of Text Highlighting Techniques. *IEEE Trans. Visualization and Computer Graphics*, 22(1):489–498, 1 2016. doi: 10.1109/TVCG.2015.2467759
- [63] L. Van Der Maaten. Accelerating t-SNE Using Tree-based Algorithms. *J. Machine Learning Research*, 15(1):3221–3245, 1 2014.
- [64] J. Visser, J. Lawrence, J. Wagemans, and C. Reed. Revisiting computational models of argument schemes: Classification, annotation, comparison. In *Proc. Computational Models of Argument*, pp. 313–324. IOS Press, 2018. doi: 10.3233/978-1-61499-906-5-313
- [65] H. Wachsmuth, N. Naderi, I. Habernal, Y. Hou, G. Hirst, I. Gurevych, and B. Stein. Argumentation Quality Assessment: Theory vs. Practice. In *Proc. Ann. Meeting of the Association for Computational Linguistics*, pp. 250–255, 2017. doi: 10.18653/v1/P17-2039
- [66] D. N. Walton. *Argumentation Schemes for Presumptive Reasoning*. L. Erlbaum Associates, 1996.
- [67] J. Wood, P. Isenberg, T. Isenberg, J. Dykes, N. Boukhefifa, and A. Slingsby. Sketchy Rendering for Information Visualization candidate depending on the first letter of their surname . *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2749–2758, 2012. doi: 10.1109/TVCG.2012.262
- [68] M. Wörner and T. Ertl. SmoothScroll: A Multi-scale, Multi-layer Slider. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, pp. 142–154. Springer, Berlin, Heidelberg, 2013.
- [69] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. In *Proc. Int. Conf. on Data Mining*, pp. 211–222. Society for Industrial and Applied Mathematics, Philadelphia, PA, 4 2010. doi: 10.1137/1.9781611972801.19
- [70] I. Zukerman. Book Review: Argumentation Machines: New Frontiers in Argumentation and Computation. *Computational Linguistics*, 31(1), 2005.