

Research Article

Andre Breitenfeld, Florian Berger, Ming-Tung Hong, Maximilian Mackeprang, and Claudia Müller-Birn*

Generating Structured Data by Nontechnical Experts in Research Settings

<https://doi.org/10.1515/icom-2018-0005>

Abstract: Semantic technologies provide meaning to information resources in the form of machine-accessible structured data. Research over the past two decades has commonly focused on tools and interfaces for technical experts, leading to various usability problems regarding users unfamiliar with the underlying technologies – so-called nontechnical experts. Existing approaches to semantic technologies consider mostly consumers of structured data and leave out the creation perspective. In this work, we focus on the usability of creating structured data from textual resources, especially the creation of relations between entities. The research was conducted in collaboration with scholars from the humanities. We review existing research on the usability of semantic technologies and the state of the art of annotation tools to identify shortcomings. Subsequently we use the knowledge gained to propose a new interaction design for the creation of relations between entities to create structured data in the subject-predicate-object form. We implemented our interaction design and conducted a user study which showed that the proposal performed well, making it a contribution to enhance the overall usability in this field. However, this research provides an example of how technically sophisticated technology needs to be “translated” to make it usable for nontechnical experts. We need to extend this perspective in the future by providing more insight into the internal functioning of semantic technologies.

Keywords: Semantic Web, Semantic Technologies, Structured Data, Usability

*Corresponding author: Claudia Müller-Birn, Freie Universität Berlin, Human-Centered Computing, Berlin, Germany, e-mail: clmb@inf.fu-berlin.de

Andre Breitenfeld, Florian Berger, Ming-Tung Hong, Maximilian Mackeprang, Freie Universität Berlin, Human-Centered Computing, Berlin, Germany, e-mails: andre.breitenfeld@gmail.com, florian.berger@fu-berlin.de, mingtung.hong@fu-berlin.de, maximilian.mackeprang@fu-berlin.de

1 Introduction

Over the past few decades, the Internet has provided an infrastructure accelerating advances in research practices across many academic disciplines. These technological advances lead to a more data-intensive research practice [19] and more interdisciplinary collaborations, allowing researchers to tackle more complex questions (e.g. [25]). This development of a more collaborative, data-intensive research practice is often referred to as *e-research*, which describes the use of digital tools and data for the distributed and collaborative production of knowledge (cp. [33]). However, a crucial impediment for e-research is the lack of a sustainable data practice [3], since it is a prerequisite for collaborative data-driven research across disciplinary boundaries. In order to combine data from different disciplines it is crucial that they are available in a structured format. Such a format would allow the mapping of different meanings in specific domain areas to other domains. This integration can be realized using the linked data principles [38]. Based on knowledge representation and semantic technologies, data can be decomposed into small entities in the form of statements to describe it (e.g. text, image). These interrelated entities can be enriched with additional information, such as context and provenance. Linked data provides mechanisms to publish structured data on the World Wide Web, where this data can then be connected and integrated in various contexts [2]. Providing data, for example, in a linked data format is one possible approach to realize the FAIR Data Principles: Findability, Accessibility, Interoperability, and Reusability [50].

Scholars in research settings can be both providers of linked data who annotate their information resources according to specific guidelines and publish them, and users of linked data who search, query or explore data from diverse sources [30]. This allows them to analyze their data at varying levels of detail by “*creating a highly effective knowledge system, particularly when integrated with other data*” [39]. In this context, machines, i.e. computational agents, are enablers that can identify, aggregate, and recombine structured data at a large scale in a meaning-

ful way. However, all these possibilities depend on the availability of structured data. Thus, data publication is fundamental to “*facilitate and simplify this ongoing process of discovery, evaluation, and reuse in downstream studies*” [50]. This is where vision meets reality. Looking at the current practice of data publication is rather sobering. The W3C,¹ for example, notes that: “*Data publication is seen as a specialist activity, not as something anyone can do, and therefore it is more centralized than expected*” [1]. Their vision is that “*individuals would publish data in much the same way that they were already publishing Web pages.*”

One approach to promote the publication of data in e-research is the automatic analysis of research articles to provide scientific statements from these articles as linked data [9]. This approach appears to be technically feasible in areas such as biomedicine, where genes and their interactions, for example, are explicitly mentioned [42]. In areas such as the humanities, relationships between entities are described more subtly and automatic approaches fail. Existing tools that allow the manual annotation of structured data are often cumbersome [20] and cannot be used by nontechnical experts without any prior knowledge of semantic technologies [11, 34]. Thus, although semantic technologies show promising capabilities, their utilization in e-research is rather low. Oldman et al. argue that this is caused by a rather technology- than user-driven integration of semantic technologies in research practice [39]. This perspective corresponds to research carried out by Karger, who states that a reason for the low adoption of semantic technologies is their lack of usability [26], which especially hinders nontechnical experts to exploit their full potential.

In the research presented here, we pursue first steps to close this gap by providing an interaction design that allows nontechnical experts to create structured data from textual resources without prior knowledge of semantic technologies. This research was conducted in close collaboration with scholars from the humanities. During the project, we have investigated scholarly research practice in terms of reading and working with textual resources [34]. This work’s contributions are: (1) to highlight existing usability challenges in the adoption of semantic technologies, (2) to review existing annotation concepts in software tools for manual annotation, (3) to propose design rationales for an interaction design for relationship annotation,

and (4) to evaluate the proposed interaction design in a user study.

In the first part of this paper, we describe findings of related work on usability challenges in the semantic web context and usability of annotation tools. Based on this research, we conducted a review of existing semantic annotation tools, the results of which are presented in Section 2. In Section 3 we propose an interaction design for annotation relation based on both the related work and the design gaps uncovered by the tool review. We implemented the design in the context of the neonion annotation tool, as described in Section 4, to create a testable artifact. Section 5 examines the usability of the design proposed based on a user study. Finally, we discuss implications of our research in Section 6.

2 Theoretical Background and Related Work

Annotation is an ubiquitous task in a wide range of applications, especially unstructured annotation, such as commenting, highlighting and tagging. The class of tools capable of creating and managing digital annotations is heterogeneous. It ranges from general to single-purpose tools, predominantly for linguistic annotation² [4].

In the following we will review literature of related usability studies on annotation tools. Afterwards, we will select and analyze annotation tools related to the research presented. As motivated before, our overarching goal is to simplify the creation of structured data from textual resources to improve the data publication practice in e-research. Therefore, we focus on software that allows the creation of semantic annotations. Criteria and attributes will be introduced to determine whether a tool should be taken into further consideration. Finally, based on the analysis of the remaining tools we will reveal differences and similarities in their conceptual model for semantic annotation. We propose a new interaction design based on the findings of this study in the subsequent sections.

2.1 Usability of Semantic Web Technologies

We will first review usability concerns in semantic technologies and then look specifically at semantic annotation tools. According to Oldman et al., two groups of users

¹ The World Wide Web Consortium (W3C) is an international community which develops Web standards. The W3C’s mission is to lead the Web to its full potential. Check out <https://www.w3.org> for more information.

² Linguistic annotation concerns the enrichment of corpora with linguistic information, e. g. part-of-speech tagging [23].

can be differentiated [39]: technical experts (e. g. research engineers and computer scientists) and users, better described as “nontechnical experts.” Nontechnical experts in this work refer to experts from a nontechnical domain, such as the history of science, using annotation tools to support their hermeneutic research process. As we will explain later, these nontechnical experts are needed to create correct statements (i. e. annotations) from textual resources. Technical experts (e. g. ontology engineers, software developers) use semantic technologies for standardizing data models, integrating data sets from different contexts or developing algorithms for automatic reasoning³ (e. g. [7, 41, 44]). Many tools have emerged from these and other use cases to support technical experts in understanding and visualizing information the semantic web exhibits. Previous research, for example, focused on developing intuitive and comprehensive representations of ontologies via visualization [29], or on automatically generating visualization from domain-specific data [14]. Users who are less technically experienced are rarely considered when developing semantic tools. Research has identified multiple reasons for the low usability of existing applications. Examples are information overload due to the complex data model or the technology involved [24], differences between the mental model of the technical and nontechnical experts [24], a low interactivity of the interaction design provided [11], and a lack of usability studies [13]. Furthermore, research that takes the user perspective into account considers nontechnical experts primarily as consumers of structured data. Thus, they address activities such as searching, querying, exploring or browsing of structured data (e. g. [11, 24, 30]). As mentioned before, users can also be producers, i. e. providers of structured data. This is especially important, since generating high quality structured data cannot be done by machines alone [6]. Users need to define meaningful statements by annotating information resources, such as texts or images. Thus, providing semantic annotation tools can allow nontechnical experts to publish structured data in the form of statements based on their research resources, given these tools can be integrated easily in existing research practice. However, this requirement seems to be too ambitious, since the usability of annotation tools has largely been absent in research and implementation [4]. Even though the topic of semantic annotation is not new in the research community, the complexity of the underlying technology

leads to a continuous reinvention of the wheel. Research projects are often terminated before user tests can be carried out. The evaluation of low fidelity prototypes such as those carried out by Hinze et al. [20], is a practice rarely seen in the area of semantic technologies.

Another exception from an evaluation perspective is the doctoral thesis of Burghardt [5]. He examined a number of annotation tools regarding their user interfaces. However, his work focuses on tools suitable for linguistic annotation which include general purpose tools. In the context of linguistic annotation, relationships are, for example, used to annotate co-references.⁴ A subordinate goal of his research was the identification of positive and negative aspects of existing interfaces. Positive aspects are considered to be best practices, whereas negative aspects point to potential usability problems. Burghardt carried out an evaluation based on a heuristic walkthrough method⁵ based on Nielsen’s usability heuristics [37]. He revealed, for example, a violation of the usability principles *error prevention*, *recognition rather than recall*, and *flexibility and efficiency of use* [5, p. 230]. The creation of relationships for co-reference annotations belonged to the set of tasks covered by the tool analysis [5, p. 98]. An example listed as unintuitive is the interface of the software “Brat.” He states that “relations between two existing annotations are created by means of click (first annotation) and release (second annotation)” [5, p. 347]. As a positive example, he recommends the interaction design of the annotation tool “Glozz.” The positive design uses a drag-and-drop interaction with visual connectors for relations and a special “annotate relations” mode. Based on this evaluation, Burghardt proposes a framework of usability patterns which provides generic solutions for annotation tools [5, p. 143]. We considered these patterns when developing our interaction design for relationship annotation. Since our goal is to improve our understanding of existing concepts of annotating relations, we review existing semantic annotation tools in more detail next.

2.2 Review of Semantic Annotation Tools

In the first part of the tool review, we explain the basic concepts of semantic annotation to motivate the criteria needed to identify appropriate tools. Subsequently, we introduce *attribute listing* [18] as our methodological basis

³ Automatic reasoning refers to the ability to generate conclusions automatically using logical techniques, such as deduction and induction from ontologies.

⁴ A co-reference describes the fact that two words or phrases refer to the same entity [47].

⁵ The Hybrid evaluation approach combines ideas from heuristic evaluation and cognitive walkthrough [43].

for analyzing the semantic annotation tools. In the third and last paragraph of this section, we summarize our insights.

2.2.1 Identifying Semantic Annotation Tools

Semantic annotation is a general term for different types of semantic enrichment. Kiryakov et al. state that a semantic annotation “*is about assigning to the entities in the text links to their semantic descriptions*” [28], Uren et al. state that a semantic annotation “*formally identifies concepts and relations between concepts in documents*” [46], and Talantikite et al. explain “*a semantic annotation is referent to an ontology*” [45]. In the following, we recap briefly the aforementioned terms to provide a shared understanding.

Structured data can state facts about any kind of thing or entity in the world. These entities can be concrete persons, places, etc. Each entity has a unique identifier (URI). Statements assert the properties of entities, in other words, an entity has a property which consists of a value. This is expressed by a triple: “Subject – Predicate – Object” (SPO). A resource (subject) has a property (predicate) with a value (object). An object can be another entity or a literal (e. g. a number or a string). Such structured data is often stored in a graph database format by using RDF.⁶ Several triples taken together form an RDF graph, whose nodes are URIs and whose arcs are properties. While assertional knowledge comprises statements about particular individuals and situations, terminological knowledge introduces the vocabulary (schema) of an application domain. RDFS (Resource Description Framework Schema), for example, is used to define the structure of the data, which allows the making of statements about classes of entities or types of relationships. The OWL (Web Ontology Language) is used to describe ontologies. This standard allows one to define semantic relationships by classes and properties, also expressed in triples. Thus, OWL adds semantics to the schema. It is assumed, for example, that the entities “Person” and “Political Party” are concepts of the terminological knowledge. The concepts given can be conceptually connected by a membership relation “is member of”. Therefore, the subject “John F. Kennedy” and the object “Democratic Party” are connected by the predicate “isMemberOf”. The order of the subject-predicate-object

triple determines the direction of the relation. OWL, for example, even allows one to indicate whether relationships are symmetric (e. g. “If A isMarriedTo B” then this implies “B isMarriedTo A”).

In summary, semantic annotations make implicit knowledge explicit by structuring unstructured content, such as texts. Consequently, a controlled vocabulary (e. g. an ontology) is used to share this data with existing knowledge bases (e. g. Wikidata [48]). These considerations lead to the first criterion for identifying suitable tools: A tool should be capable of creating semantic annotations by providing an annotation schema for relationships based on a controlled vocabulary (C.1). Secondly, we focus on manual annotation because of the project’s context in humanities research (C.2). The third criterion requires the annotation of textual resources such as plain or formatted text, including web pages (C.3). The last criterion constitutes the possibility of checking a tool either by installing it from source code or by using an online or binary distribution, because testing an interaction design based on screenshots is only possible to a limited extent (C.4). Thus, we included only software in the tool review that meets all four criteria.

Based on our literature review, we identified 24 tools (cp. Table 1).⁷ Each tool was checked individually. We checked the defined selection criteria by studying the user’s manual, feature description, and web page. All tools support the annotation of named entities,⁸ but only about half of them considered computer-based support for annotating relationships. As shown in Table 1, we selected 12 tools for the subsequent detailed review.

2.2.2 Attribute-Based Tool Analysis

To recap, our goal is to propose a new interaction approach for facilitating annotating relationships (following a triple format) for providing structured data on the Web. Our first overview of the tools revealed that the interaction design

⁶ RDF (Resource Description Framework) is a graph data standard from the W3C that encodes directed graphs with different types of edges. It supports the storage of contextual information.

⁷ We used primarily three sources for identifying these tools: (1) A doctoral thesis on the usability of linguistic annotation tools [5], (2) a survey of general purpose tools for manual annotation of semantic facts in biomedical literature [36], and (3) Google Scholar. Additional information of the tool selection criteria and all the results of the review are provided in more detail in a technical report available at http://edocs.fu-berlin.de/docs/receive/FUDOCs_document_000000027360

⁸ Named Entity Extraction refers to the task of identifying and classifying named entities in text. Exemplarily, identifying named entities of type person, location and date in a text.

Table 1: Overview on annotation tools identified from literature and the selection criteria applied.

Tool	Web	C.1	C.2	C.3	C.4	Source
@Note	https://omictools.com/note-tool	×	×	✓	✓	[36]
Analec	http://lattice.cnrs.fr/Analec,68	✓	✓	✓	✓	[5]
Argo	http://argo.nactem.ac.uk	×	×	✓	✓	[36]
Bionotate	http://bionotate.sourceforge.net	✓	✓	✓	✓	[36]
Brat	http://brat.nlplab.org	✓	✓	✓	✓	[5]
Callisto	http://mitre.github.io/callisto	×	×	✓	✓	[36]
CATMA	http://catma.de	×	×	✓	✓	[5]
CorpusTool	http://corpus-tool.com	×	×	✓	✓	[5]
Djangology	http://djangology.sourceforge.net	×	×	✓	✓	[36]
DOMEO	http://annotationframework.org	×	×	✓	✓	[8]
FLERSA	–	✓	✓	✓	×	[35]
GATE	https://gate.ac.uk	✓	✓	✓	✓	[5]
Glozz	http://glozz.free.fr	✓	✓	✓	✓	[5]
Inforex	http://inforex.clarin-pl.eu	✓	✓	✓	✓	[31]
Knowtator	http://knowtator.sourceforge.net	✓	✓	✓	✓	[5]
MMA2	http://mma2.sourceforge.net	✓	✓	✓	✓	[5]
MyMiner	http://myminer.arimi.monash.edu.au	×	×	✓	✓	[36]
Pundit	http://thepundit.it	✓	✓	✓	✓	[16]
RDFaCE	http://rdface.aksw.org	×	×	✓	✓	[27]
Semantator	http://informatics.mayo.edu/cntro	×	×	✓	×	[36]
Vogon	http://gobtan.sourceforge.net	✓	✓	✓	✓	[12]
WebAnno	http://webanno.github.io	✓	✓	✓	✓	[5]
WordFreak	http://wordfreak.sourceforge.net	✓	×	✓	✓	[5]
Xcon Suite	http://geniaproject.org	✓	✓	✓	✓	[36]

to create relation annotations is not a standardized process. The review of the interaction designs of the semantic annotation tools proceeds in two steps: (1) a classification of each interaction design by an interaction pattern and (2) a characterization of research-related attributes. The first step represents a review on a higher-level. We classified the chosen interaction pattern for relationship annotation of each tool. In the second step, we reviewed the tools selected on a more detailed level. We employ the analytical approach *attribute listing* as the methodological basis [18]. Crawford [10], for example, developed a technique for designing and re-designing a solution by decomposing an existing solution into attributes. Attributes are features or properties of a solution and represent unbiased feedback. The design of alternative solutions is accomplished by modifying these attributes.

We organized the attributes into three groups: *feature-specific*, *interface-specific* and *interaction-specific* attributes. A discrete value represents each attribute. A dichotomous, i. e. a nominal scale with two levels, was used to describe whether a specific feature is present in one way or another. Binary scales were used to decide whether a function exists generally.

Feature-specific attributes concern the capability of a tool. We checked whether a tool allows the annotation of entities (A.1) and the annotation of a relation (A.2). Since

provenance information is essential for the trustworthiness of data [17], we investigated whether an annotation includes any provenance information, such as creator and annotation date (A.3). The last attribute (A.4) in this category relates to the origin of the terminological terms, thus, the use of external vocabularies (e. g. Wikidata [48]).

The group of *interface-specific* attributes concern structural and behavioral features of the interface for the creation of annotations. Structural features consider the accessibility of the interface. At first, we analyzed whether an interface offers a particular mode to create relationships, which means it is not possible to create a relationship without accessing this mode (A.5), and tested if this interface is separated from or integrated into the text (A.6). Behavioral features relate to the physical size and the location of the interface for creating relation annotations, depending on the position of the entities of the relationship. The first aspect considers the way the interface allocates the space on the screen. A screen space can be allocated *constantly* independent of the entities or *proportionally*, for example, in relation to the entities (A.7). The second aspect analyzes the spatial positioning of the interface. An interface can either be placed at a *fixed* or *relative* position on the screen, for example, relative to the entities in the text (A.8). As the number of terminological concepts increases, the combination of possible relations

Table 2: Review results by feature-, interface- and interaction-specific attributes. All attributes have a binary scale, except A.7 (Constant / Proportional), A.8 (Fixed / Relative) and A.14 (number of clicks).

		Analect	BioNotate	Brat	GATE	Glozz	Inforex	Knowtator	MMA2	Pundit	Vogon	WebAnno	XConc
Feature-specific attributes													
A.1	Annotation includes entity annotation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
A.2	Annotation includes relationship annotation	✓	✓	×	×	✓	×	×	×	×	✓	×	×
A.3	Annotation includes provenance information	×	×	×	×	✓	×	×	×	✓	✓	✓	×
A.4	Relationships from ontological properties	×	×	×	✓	×	×	✓	×	✓	✓	×	✓
Interface-specific attributes													
A.5	Interface offers a special mode for relations	×	×	×	✓	✓	✓	×	×	×	✓	×	✓
A.6	Interface is separated from the text	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓
A.7	Screen space allocation of the interface	<i>C</i>	<i>P</i>	<i>C</i>	<i>C</i>	<i>P</i>	<i>C</i>	<i>C</i>	<i>P</i>	<i>C</i>	<i>C</i>	<i>C</i>	<i>C</i>
A.8	Spatial positioning of the interface	<i>F</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>R</i>	<i>F</i>	<i>F</i>	<i>R</i>	<i>F</i>	<i>F</i>	<i>F</i>	<i>R</i>
A.9	Interface offers user assistance to avoid errors	×	✓	×	×	×	✓	×	×	×	×	×	×
A.10	Interface uses the subject-predicate-object paradigm	✓	×	✓	×	×	✓	×	×	✓	✓	✓	×
A.11	Interface introduces an abstraction layer	×	✓	✓	×	✓	×	×	×	×	×	✓	×
Interaction-specific attributes													
A.12	Interaction involves the visual anchors	✓	×	✓	×	✓	✓	×	✓	✓	✓	✓	✓
A.13	Interaction order effects relation direction	×	×	✓	✓	✓	✓	×	✓	×	×	✓	✓
A.14	Interaction effort in mouse clicks	5	2	3	6	4	5	10	3	8	8	3	4

increases as well. Thus, a user needs to memorize all possible relations for each pair of concepts. This mental effort can be reduced with the help of user assistance (A.9). Furthermore, abstraction is an important factor when it comes to triple annotation. We checked whether a tool uses the SPO paradigm in the interface directly (A.10) and tested whether a tool provides an abstraction layer for relations (A.11) or not.

The last group of *interaction-specific* attributes focuses on the interaction design. The first attribute concerns the role of the entity annotation when a relation is created. We tested whether the interaction involves visual anchors for the entities or not (A.12). We then analyzed the effect of the ordering on the direction of the relationship, in other words, if the order of interaction determines the direction implicitly (A.13). Finally, we assessed the minimum number of clicks necessary to create a relationship (A.14).

The feature-, interface- and interaction-specific perspective led to a set of 14 attributes which are used to investigate the 12 tools in more detail. Our insights are summarized in the following section.

2.2.3 Tool Review Findings

Table 2 summarizes the results of the review organized by tool and attribute. All tools included the entity annotation, while only a minority considered relationship annotation.

The predominant model for annotating a text was that a user can explicitly define the entities by using the underlying vocabulary, whereas the relations are implicitly derived from the context of the text. Only four tools provided provenance information on the annotation.

Furthermore, the results show that 7 out of the 12 tools used the SPO paradigm as the basis for the user interface and 4 only introduced a kind of abstraction. When the SPO paradigm was chosen, the interaction pattern most often used was a slot-filling approach (5 out of 7): Users have to select subject, predicate and object from a drop-down menu to create an annotation. This shows that the focus on technical users, as described by literature, is apparent in the absence of an abstraction layer over the data representation of most tools (cp. [24, 11]).

Almost all tools separate the text and the interface to create a relation (e. g. the Triple Composer in Pundit). The separated interface appears mainly at a fixed position on the screen. This creates a situation where the cursor is constantly moving between the different panels to reach a specific feature and it increases the mental effort of users when connecting the text with the triple structure. Consequently, the mouse movement is subjectively high in these tools. Regarding the click effort, the review pointed out cumbersome approaches, such as Knowtator, Vogon and Pundit. In addition, some tools have a screen-space inefficient visualization, such as Brat, WebAnno and XConc Suite. Another finding was that most of the tools

had no means of preventing annotating relations that conflict with the underlying semantic model. Only two of the tools leveraged constraints given by the relationship definition to prevent potential user errors.

The review detected two major design gaps for enabling an annotation practice for nontechnical experts. The first design gap concerns an adequate abstraction level. A missing abstraction layer forces the users to familiarize themselves with the underlying data layer and concepts used in semantic technologies. Only four tools provide an abstraction layer while the majority of tools uses the SPO paradigm or a similar approach. The second design gap refers to the assistance of users. Only two tools provide precautions to prevent errors. This is a major issue, for example, when considering the DBpedia ontology, where the concept *person* has more than 250 possible properties⁹ it might be challenging for a nontechnical user to find the correct and most meaningful property. Thus, the importance of a user assistance needs to be emphasized. In the following, we use these insights to propose a new interaction for relation annotation.

3 Interaction Design for Relation Annotation

One key factor in the lack of adoption of semantic technologies is a missing focus on nontechnical experts, as seen in the related work (Section 2.1). Furthermore, this became apparent, in the review of Semantic Annotation Tools (Section 2.2) in two design gaps: Firstly, the missing abstraction layer, forcing the users of annotation software to understand the technical details of the SPO paradigm, and secondly, the lack of user assistance. Based on both the usability patterns analyzed in related work and the design gaps uncovered by our review, we developed an interaction design providing an abstraction for relationship annotation. The goal of the design subsequently described is to facilitate the annotation of relationships between entities in a text. The interaction design is based on the following scenario: The user has already annotated entities in a text and wants to annotate relationships between the entities. The design uses a connection-centric abstraction approach and is guided by four underlying design rationales. These rationales are introduced in the following.

⁹ The DBpedia Ontology (<http://wiki.dbpedia.org/services-resources/ontology>) is a cross-domain ontology, which has been manually created based on the infoboxes within Wikipedia. The concept *person* can be found at <http://dbpedia.org/ontology/Person>

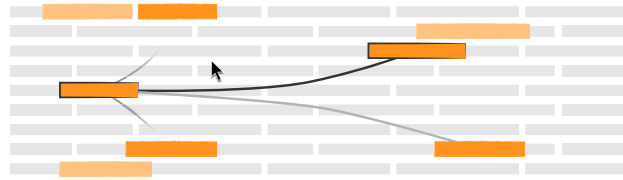


Figure 1: Connecting lines between entities.

3.1 Expose the User's Options

Conceptually, the solution proposed draws connections and suggests relations where appropriate. Only connections which are meaningful with respect to the underlying vocabulary are visible for users to account for the design gap identified in user assistance. Connections are hidden in the initial state of the interface. Clicking on an entity enters the relation mode. The annotation selected will be visually connected to all meaningful entities (see Figure 1). Entities which are not meaningful fade. Pratt et al. state that animation attracts the user's visual attention. They found that objects involving animate motion were noticed more quickly than objects that did not [40]. Using this principle, connection lines are animated to expand from the selected annotation to the endpoints to make users aware of their options. Connecting lines are curved for aesthetic reasons. Furthermore, connections are selected by proximity. The connection with the smallest Euclidean distance to the cursor is selected (see highlighted connection in Figure 1). By selecting a connection, the subject and object of the relation are determined implicitly by the underlying vocabulary.

3.2 Focus the User's Intentions

In certain situations, the number of visible connections might be too high, which could lead to an information overload [24]. We decided to minimize the cognitive load by limiting the number of visible connections to a visual range. This range is aligned to the user's area of interest. The area of interest corresponds to a user's gaze.¹⁰ Figure 2 illustrates how we used the cursor to construct the visual range. The center of the annotation selected and the cursor form a direction vector " v ". The angle α denotes the opening angle of the visual range. It is set to a fixed value of 10° .

¹⁰ Huang et al. conducted a study on the gaze-cursor alignment. They concluded that there is a correlation between gaze and cursor position while the user examines and reads a web page. Furthermore, gaze-cursor alignment is particularly high if the user tries to target a visual element to perform an action [22].

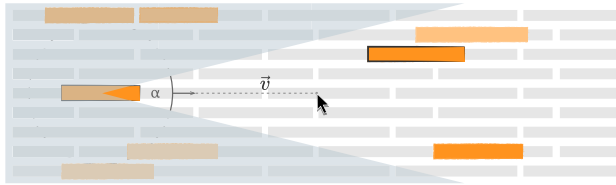


Figure 2: Reducing number of visible annotations by a limited field of view.

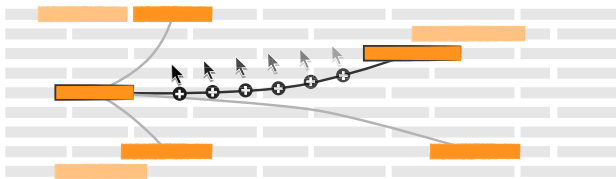


Figure 3: Movement of the Runner over time, when creating a new relation.

Only connections whose annotation anchors are inside of the field of view become fully visible. Connections whose annotations are outside are shortened (Figure 1). Existing relations are always visible to the user.

3.3 Accompany the User

We introduce the concept of a Runner that is a visual element which follows the user's cursor along the selected connection path (cf. Figure 3). Fitts's law motivated the usage of a Runner, because selecting small objects takes a longer time [32]. Connecting lines are thin objects and rather difficult to click on. In the light of Fitts's law, the Runner serves as a larger clicking target. The Runner has two states that decode whether a relationship exists on the connection selected or not. A plus icon indicates the absence of a relation and a pen icon indicates a present relation that can be modified. Clicking the icon on the Runner opens the statement menu, which is described as follows.

3.4 Speak the User's Language

The statement menu is a context menu. It represents the last step to complete the definition of the relationship. We applied the principle of least astonishment to help nontechnical experts in understanding their options. Therefore the statement is presented as a sentence in natural language. These statements are populated and grammatically composed based on the underlying vocabulary. Thus, only permitted relations between the entities selected are provided (preventing errors in created relations). All infor-

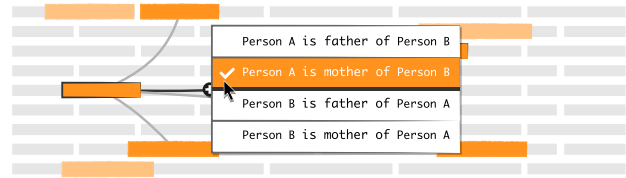


Figure 4: Statement menu on top of the Runner with annotation options with bidirectional support in natural language.

mation about the entities involved and type and direction of the resulting relation are expressed as one sentence. Since it must be assumed that a relation already exists, the statement menu provides features to change or delete existing relations. A checkmark next to a statement denotes the existence of a relationship. In the case where a relation already exists, the statement menu additionally shows the creator of the relation. As described before, relationships can be symmetric. We therefore introduced a bidirectional assistance (cp. Figure 4). Because of the bidirectional assistance, users do not need to consider the order of their interactions.

In order to evaluate the efficiency of the aforementioned design rationales, we implemented the proposed solution in a relationship extraction mechanism embedded in the semantic annotation software *neonion* which is explained in the following in further detail.

4 Implementation

In the last section, we described an interaction design tailored to address the shortcomings of related approaches regarding nontechnical users. In order to evaluate the design, we implemented it as an extension of an existing semantic annotation tool. The following section outlines the technical details of this software.

4.1 Semantic Annotation Software

neonion is a user-centered web application for the collaborative annotation of text documents. It is being developed at the Human-Centered Computing Group at Freie Universität Berlin and is available as open source software.¹¹

The core of the tool is an intuitive browser-based user interface to add annotations to texts manually and man-

¹¹ For more information, please check out the website <http://neonion.org/>. The source code is available at <https://github.com/FUB-HCC/neonion> and a demo version <https://demo.neonion.imp.fu-berlin.de>

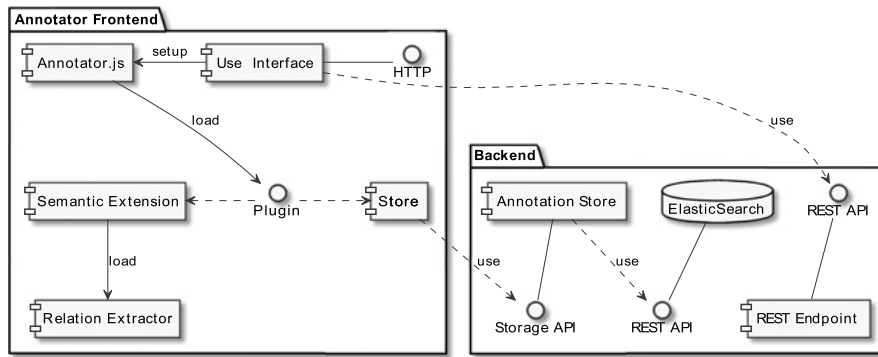


Figure 5: System architecture of the neonion semantic annotation software.

age existing annotations. Highlighting parts of texts and commenting on them makes use of familiar user interface concepts. The semantic annotation capabilities of neonion feature a flexible knowledge model based on an extensible vocabulary, as well as additional metadata, such as timestamp, owner, and the annotation set, i. e. vocabulary, used.

The user interface provides a document tool which allows for upload, conversion and management of text files. User and group management and a permission system for “public,” “group” and “private” contexts are included. Users can work collaboratively in teams and share documents and annotations.

Internally, neonion is strictly separated into a front-end and a back-end part. The front-end is a browser-based web application, consisting of a view made up of HTML and CSS, and the controlling code written in JavaScript. The component responsible for the actual annotation interface is based on an existing software library “Annotator.”¹² It provides a general infrastructure to create annotations on top of HTML markup. The software is open-source and also extensible via plugins. The neonion front-end deploys a custom written Annotator plugin to handle annotation semantics and to communicate with the annotation store in the back-end.

The back-end of neonion is a GUI-less server application which uses application protocol interfaces (API) to communicate with the front-end. It is built upon Django,¹³ a high-level Web framework written in the Python programming language. Django cares for the JSON HTTP interfaces, the mapping of URIs to handler components and the storage of documents and user, group and permission data.

The actual semantic annotation data is being handed over by Django to “Elasticsearch,” a NOSQL database ap-

plication specialized in the storage, indexing, search and retrieval of schema-free JSON data sets.¹⁴ It is based on the open-source Apache Lucene software¹⁵ and written in the Java programming language. Elasticsearch allows for very fast and efficient persistent saving and loading of annotations, which are being represented as JSON data sets following the W3C Open Annotation Data Model specification.¹⁶ The program enables the connecting of annotations with Linked Data sources, e. g. Wikidata.¹⁷ There is also an experimental SPARQL interface.

As outlined, front-end and back-end communicate via network APIs to perform their tasks. When the user opens a new document, it is sent to and stored in the back-end. The front-end presents the document and offers visual tools for text fragment annotation. When the user creates annotations, they are sent to the back end, which stores them in the Elastic Search database. When opening an existing document, the front-end queries the back-end for all related data, such as the document itself, the related annotations and, subsequently, renders a view for the user to interact with.

While the neonion front-end has been developed for text document annotation only, the back-end is designed in such a way that it can be used for annotating arbitrary data, as long as these can be described using established standards, i. e. as long the target an annotation refers to can be defined unambiguously. The Human-Centered Computing group is currently working on making neonion usable to annotate three-dimensional object models [21].

Figure 5 illustrates the technical architecture behind the neonion software.

¹² For more information please check <http://annotatorjs.org/>

¹³ <https://www.djangoproject.com/>

¹⁴ <https://www.elastic.co/products/elasticsearch>

¹⁵ <https://lucene.apache.org/>

¹⁶ For more information please check <http://www.openannotation.org/spec/core/>

¹⁷ <http://www.wikidata.org>

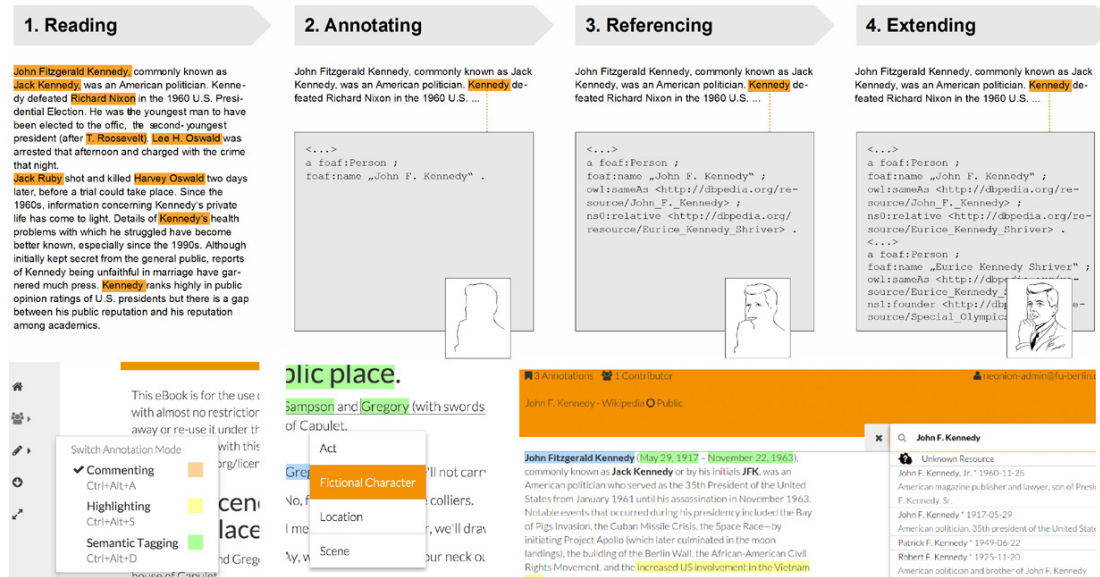


Figure 6: The general concept of the semantic annotation process in neonion differentiated into four activities from a data perspective (upper row) and a user interface perspective (lower row).

4.2 Annotation User Interface

The neonion user interface was designed in an iterative fashion, incorporating end user feedback in each step [34]. In the current version, users can select a part of the text for highlighting, adding comments that refer to the highlighted text, and create semantic annotations by linking the selected part of the text to a vocabulary. The general process of annotation is shown in Figure 6, alongside the according operations on the data. Users can annotate selected words or parts of the text with predetermined terms—so-called “concepts” that stem from the vocabulary. A user may annotate a name while reading the text, for example, “John F. Kennedy.” The user connects this name to the concept “Person” first, and then links it to a specific instance, in our example “John F. Kennedy.” This structured vocabulary resource might reference the Wikidata entry “Q9696.” Thus, more information about the person “John F. Kennedy” can be retrieved, or Wikidata can be extended by information in the text instead. This is not only important when searching for existing resources or linking to new ones, but also for ensuring the re-usability of annotations.

Users can reuse all types of annotations, either by reusing selected annotations or by exporting all annotations. The export is adaptable and supports a variety of formats, which enables the sharing of annotations between different annotation tools.

We integrated the proposed design by extending the Open Annotation Data Model with a *RelationshipMention*

John Fitzgerald Kennedy (May 29, 1917 – November 22, 1963), commonly known as Jack Kennedy or by his initials JFK, was an American politician who served as the 35th President of the United States from January 1961 until his assassination in November 1963. Notable events that occurred during his presidency included the Bay of Pigs Invasion, the Cuban Missile Crisis, the Space Race—by initiating Project Apollo (which later culminated in the moon landings), the building of the Berlin Wall, the African-American Civil Rights Movement, and the increased US involvement in the Vietnam War.

Figure 7: Screenshot of the proposed design implementation in the context of the neonion software.

and integrating the interaction design as an “Annotator” Plugin. Figure 7 shows how the proposed design was finally implemented in the neonion GUI. This implementation was used in the following evaluation.

5 Evaluation of the Proposed Design

After implementing the interaction design in the context of neonion, we conducted a within-subject usability study to evaluate it. The study was designed to reflect the requirements defined by both related work and the tool review conducted. Its goal was to compare the approach to state-of-the-art tools in terms of *learnability*, *effectiveness* and *satisfaction*.

The participants in the study were researchers and students from local universities and research institutes. They were recruited by mailing list (13 male, 3 female). One participant was excluded from the study due to technical

problems during the test session. The participants received no compensation for their participation.

We provided a use case during the study in which participants were requested to generate structured data from textual resources. We told each participant that the task is to select a tool that is most appropriate for generating structured data from text for nontechnical experts. Even though most participants (13) indicated that annotating resources is an important part of their daily work, they had little knowledge of semantic annotation tools. We conducted a comparative usability study based on three tools to evaluate whether our proposed design outperforms existing approaches or not. One tool (T1) was representative of the slot-filling interaction paradigm. We chose one of the tools providing an abstraction layer for the second tool (T2). The third tool (T3) was our own implementation.¹⁸ A within-subjects design was employed following the recommendation of literature on competitive studies [15]. The order of the tools' presentation was randomized to mitigate familiarization effects. The study was originally conducted in German. The questions and qualitative results presented subsequently were carefully translated for publication.

5.1 Methodology

We focused our evaluation on the following criteria in the usability study: learnability ($C_{\text{learnability}}$), satisfaction ($C_{\text{satisfaction}}$) and effectiveness ($C_{\text{effectiveness}}$) to consider especially the requirements revealed in previous research [11, 13, 24].

5.1.1 Tasks

We divided the study into two phases. In the first phase, i. e. the learning phase (P_{learning}), we assessed the learnability of the interfaces provided. After the first phase, we explained all interfaces to the participants and, thus, prepared the second phase, i. e. the working phase (P_{working}). The satisfaction and effectiveness have been evaluated in both phases. Learnability and satisfaction were determined by a questionnaire (Q1: “How easy was it to learn the tool after the first-time use?”; Q2: “How much do you

like the tools after the first-time use?”; Q3: “How much do you like the tools after you’ve worked with them?”). Because we carried out a comparative test design, test participants were asked to order the tools from “easy to use” through “neutral” to “difficult to use”. Effectiveness was measured by task completion. We encouraged participants to verbalize their thoughts while using the tools (“Thinking Aloud”) to receive qualitative feedback. Since the focus of the solution designed is on the annotation of relationships, we tested the creation of relationships only. Therefore, participants received a text where entities (subjects, objects) were already annotated in each tool. One relationship had to be created in the learning phase. The task had to be completed without prior explanation of the tools. We set a time limit of three minutes for this task and tracked completion of the task to evaluate the effectiveness of the design. After completing the task, participants were asked to rank the tools in terms of their learnability and satisfaction. Afterwards, they received a detailed explanation of the tool’s functionality. The working phase consisted of six different annotation creation tasks. The difficulty of the tasks was similar except for the last task where users had to create a symmetric relationship. The time limit for this phase was set at five minutes. After completing the working phase, the participants were asked to fill out another questionnaire.

5.2 Results

Table 3 shows the results of our study for each phase and tool. Each row shows the rankings picked by the relative amount of people for each tool (e. g. the first cell shows that 27 % of the participants chose T1 as the best ranking tool in terms of ease of use). Furthermore, we added the task completion rates for a concise reference to all results. We explain these results in more detail in the following.

After the learning phase, 73 % of the participants found the proposed interaction design was the easiest in terms of learnability (Q1: as shown in Figure 8) and again, 73 % of the participants found the proposed interaction design most satisfactory (Q2: as shown in Figure 9) ($p = 0.0017$). The task completion rate for both, the proposed design and the slot-filling design (T1) was very high, as opposed to the point-and-click design provided by T3, where only 53 % of the testers could to complete the task (see Figure 11).

All participants completed all tasks within the given time limit (100 % task completion rate) in the working phase. Even though user satisfaction is still the highest for the proposed design, the difference in the results is

¹⁸ These tools were selected because of the results of the in-depth tool review described in Section 2.2. Further selection criteria were binary / source distribution availability, text layout similarity and the availability of an English localization. T1 is the tool Pundit [16] and T2 is the tool Glozz [49].

Table 3: The percentage of ratings for each tool (T1, T2, T3) is provided ($n = 15$). Each rank position represents the choices of all participants, for example, 20 % of the participants ranked T1 as best in terms of satisfaction (Q3), while 20 % ranked T2 best and 60 % ranked T3 as best. * TC refers to the task completion rate for each phase and tool.

Phase	Criteria	Tool Rank	T1 (slot filling)			T2 (point-and-click)			T3 (proposed design)		
			1 st	2 nd	3 rd	1 st	2 nd	3 rd	1 st	2 nd	3 rd
P_{learning}	$C_{\text{learnability}}$	Q1	27 %	67 %	7 %	0 %	7 %	93 %	73 %	27 %	0 %
P_{learning}	$C_{\text{satisfaction}}$	Q2	26 %	53 %	20 %	0 %	20 %	80 %	73 %	27 %	0 %
P_{learning}	$C_{\text{effectiveness}}$	TC*	92 %			53 %			100 %		
P_{working}	$C_{\text{satisfaction}}$	Q3	20 %	60 %	20 %	20 %	0 %	80 %	60 %	40 %	0 %
P_{working}	$C_{\text{effectiveness}}$	TC*	100 %			100 %			100 %		

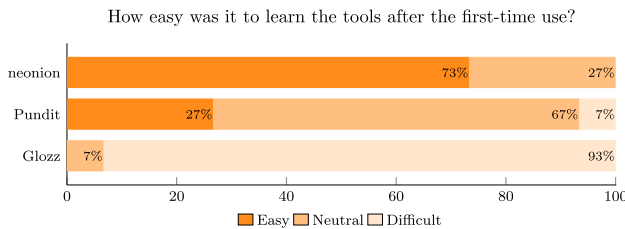


Figure 8: Summarized answers to Q1 (Learnability) after the learning phase.

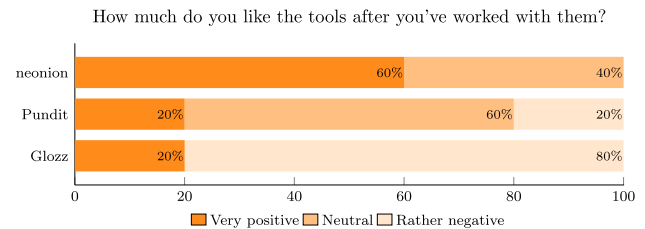


Figure 10: Summarized answers to Q3 (Satisfaction) after the working phase.

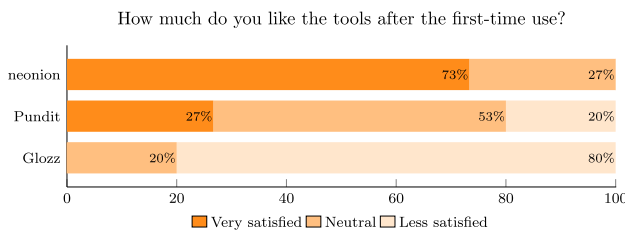


Figure 9: Summarized answer to Q2 (Satisfaction) after the learning phase.

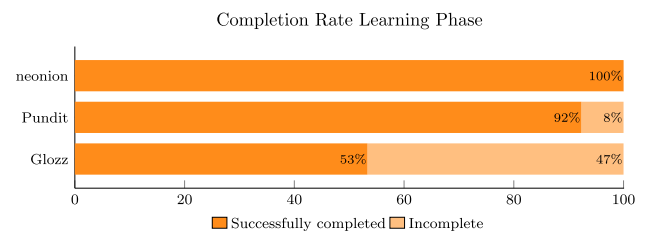


Figure 11: Task completion rates for the learning phase.

smaller. The proposed interaction design was most satisfactory for 60 % of the test participants ($p = 0.03$), but 20 % of the participants were highly satisfied for each of the other two tools (as shown in Figure 10).¹⁹

5.3 Discussion of Results

The completion rate and the high ratings for both satisfaction and learnability in the questionnaire of the learning phase suggest that the proposed design is more intuitive and easier to learn than the other two designs. The proposed design provides an interface for nontechnical experts to build up their mental models. It bridges the gap of mental models between technical and nontechni-

cal experts as proposed by Jameson [24]. We conjecture that the proposed design provides more assistance to the user to create structured data by annotating text semantically. The assistance hides the complexity of semantic technology involved successfully, as suggested by Dadzie et al. [11].

However, the results are less conclusive, after the working phase. The trade-off between learnability and effectiveness of a software system especially could not be evaluated due to the 100 % completion rate for all three tools. One reason for the high completion rate might be the low difficulty of the task provided. The text utilized in the user study was rather simple and the wording in the task was quite similar to the concepts of the vocabulary. Future work needs to evaluate the effectiveness and efficiency for more complex tasks.

We suppose that the lack of any immediate feedback in the proposed design after creating a relation is a potential source for confusion. The change of the Runner icon

¹⁹ p-values were calculated using the null-hypothesis that participants filled out questionnaires randomly.

from a plus-icon to a pen-icon does not provide enough information about whether a relationship was saved or not. The thinking aloud protocols revealed that multiple participants were confused by this fact (P3, P4, P6). Moreover, a further analysis of the thinking aloud protocols showed that the bidirectional assistance was also not easy for participants to use. One participant (P2), for example, accidentally added a relation by mixing up the direction of the statement. Instead of annotating “A was killed by B”, the person created the annotation “B was killed by A”. Even though participants were encouraged to read all statements provided by the Runner carefully, it did not prevent user errors. Further improvement of the presentation of the possible statements regarding the Runner are needed.

Another unexpected result was the low task completion rate during the learning-phase for T2 as a representative of the group of tools providing an abstraction. One possible explanation for this outcome is that the creation of relationships is not the default mode used in the tool. Qualitative feedback by multiple participants (P1, P3) indicates that users did not know how to switch into the relation-creation mode without an explanation.

5.4 Limitations of the Study

The usability evaluation was subject to some limitations. One is the selection of the tasks: The evaluation was focused on annotating relations, therefore, entities were already annotated in each tool. Thus, even though the usability of annotating relationships was considered as satisfactory and easy-to-learn, the results could be affected by annotating entities as well as relationships. The vocabularies defined in the tools were prepared for the user tasks and contained only the useful semantic relationships of classes and properties. A larger vocabulary (e. g. DBpedia) would be used in a more realistic scenario. Consequently, the importance of the proposed user assistance grows. We could only install a subset of tools in our test setup; therefore, we had to limit our study to one representative tool for each interaction paradigm. This limits the generalization of our study results.

6 Conclusion and Future Work

Many research projects have been initiated to provide researchers with appropriate infrastructure for carrying out e-research over the last decade. Only rarely has research looked at the contexts and conditions needed for conducting data publication practices in this context. How-

ever, sustainable data publication is an impediment in the e-research context for enabling the vision of interdisciplinary and collaborative data projects. One approach for realizing such data practice is to provide existing research results as linked data. Using linked data, i. e. semantic technologies, is especially challenging in contexts where nontechnical experts are needed to provide their data. In areas such as biomedicine, automatic approaches for extracting structured data are feasible, for example, obtaining explicitly mentioned genes and their interactions from texts. In areas such as the humanities, relationships between entities are described more subtly and existing automatic approaches fail. Thus, approaches are needed that support nontechnical experts in annotating textual resources.

In this paper, we proposed a new interaction design that allows nontechnical experts to create structured data from text by means of semantic annotation. Based on existing research and a tool review, we derived design rationales for a new interaction design that had been realized and evaluated in a software tool. Our results indicate that the task of creating relationships between semantic annotations can be improved in terms of learnability and satisfaction in comparison to state-of-the-art semantic annotation tools by using the proposed interaction design. This suggests that a more user-centric approach to content creation in the context of semantic technologies and linked data could increase the adoption by nontechnical experts and, therefore, might expand the overall use in both research and commercial applications. However, due to the highly specialized area of semantic relations, our design and evaluation only supports a small set of content generation tasks in the area of the Semantic Web.

Future work should expand on this by looking at other tasks and the integration of different approaches. In order to further improve the user assistance in semantic annotation processes, a possible approach would be the development of a recommender system to suggest concepts and relationships deemed most useful for the user. Additionally, in order to increase knowledge about semantic technologies, further enhancements could experiment with the integration of interactive tutorials, illustrating available annotation terms and iteratively conveying the structure of the underlying vocabulary to the user.

The project presented is based on an interdisciplinary collaboration between scholars from the humanities and researchers of computer science. It has been the goal of both partners throughout the project to critically reflect on the findings in both fields of research. The software itself caused a special tension in this collaboration. From a computer science perspective, the software is a research

artifact, designed to understand scholarly annotation processes. From the perspective of humanists, the software is a technical good that needs certain stability in order to be used effectively. This tension should be taken into account when conducting such interdisciplinary collaborations.

Acknowledgment: We would like to express our very great appreciation to all persons that contributed to the neonion software: Lukas Benedix, Tina Klüwer and Alexa Schlegel. The research presented here was only possible because of their work.

Funding: We gratefully acknowledge the financial support of the Elsa-Neumann-Stipend of the state of Berlin, of the German Research Foundation (Deutsche Forschungsgemeinschaft) with the Cluster of Excellence “Image Knowledge Gestaltung: An Interdisciplinary Laboratory” (EXC 1027), and the Federal Ministry of Education and Research of Germany (Bundesministerium für Bildung und Forschung) in the framework of “Ideas to Market” (project number 03IO1617).

References

- [1] W3C DATA ACTIVITY building the web of data. <https://www.w3.org/2013/data/>.
- [2] T. Berners-Lee. Linked Data. <https://www.w3.org/DesignIssues/LinkedData.html>.
- [3] C. L. Borgman. *Big Data, Little Data, No Data*. MIT Press, 2015.
- [4] M. Burghardt. Usability recommendations for annotation tools. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 104–112, Jeju, Republic of Korea, July 2012. Association for Computational Linguistics.
- [5] M. Burghardt. Engineering annotation usability – toward usability patterns for linguistic annotation tools, September 2014. URL <http://epub.uni-regensburg.de/30768/>. Pattern wiki: <http://www.annotation-usability.net>.
- [6] J.-P. Cahier and M. Zacklad. Socio-semantic web applications: towards a methodology based on the theory of the communities of action. *resource*, 8 (400), 2004.
- [7] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. In *Proc. of the WWW*, pages 74–83. ACM, 2004.
- [8] P. Ciccarese, M. Ocana, and T. Clark. Open semantic annotation of scientific publications using DOME0. *J. Biomedical Semantics*, 3 (S-1): S1, 2012a. URL <http://www.jbiomedsem.com/content/3/S1/S1>.
- [9] P. Ciccarese, M. Ocana, and T. Clark. Open semantic annotation of scientific publications using DOME0. *Journal of Biomedical Semantics*, 3 (Suppl 1): S1, 2012b. ISSN 2041-1480. doi:10.1186/2041-1480-3-S1-S1. URL <http://www.jbiomedsem.com/supplements/3/S1/S1>.
- [10] R. Crawford. *The Techniques of Creative Thinking: How to Use Your Ideas to Achieve Success*. Hawthorn Books, 1954. URL <https://books.google.co.uk/books?id=BCAsQAAMAAJ>.
- [11] A. S. Dadzie, M. Rowe, and D. Petrelli. Hide the stack: Toward usable linked data. In *Lecture Notes in Computer Science*, pages 93–107, 2011.
- [12] J. Damerow, B. E. Peirson, and M. D. Laubichler. Don’t panic! a research system for network-based digital history of science. In *Future of Historical Network Research Conference*, 2013.
- [13] P. Di Maio. Toward global user models for semantic technologies: Emergent perspectives. In *Proc. of ASWC*, volume 8, pages 141–152, 2008.
- [14] O. Gilson, N. Silva, P. W. Grant, and M. Chen. From web data to visualization via ontology mapping. In *Computer Graphics Forum*, volume 27, pages 959–966, 2008.
- [15] E. Goodman, M. Kuniavsky, and A. Moed. Observing the user experience: A practitioner’s guide to user research. *IEEE TPC*, 56 (3): 260–261, 2013.
- [16] M. Grassi, C. Morbidoni, M. Nucci, S. Fonda, and F. Piazza. Pundit: Augmenting web contents with semantics. *Literary and Linguistic Computing*, 28 (4): 640–659, 2013.
- [17] O. Hartig. Provenance information in the web of data, 2009.
- [18] S. R. Herring, B. R. Jones, and B. P. Bailey. Idea generation techniques among creative professionals. In *System Sciences, 2009. HICSS ’09. 42nd Hawaii International Conference on*, pages 1–10, Jan 2009. doi:10.1109/HICSS.2009.241.
- [19] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [20] A. Hinze, R. Heese, M. Luczak-Rösch, and A. Paschke. Semantic enrichment by non-experts: usability of manual annotation tools. In *ISWC*, pages 165–181. Springer, 2012.
- [21] A. Hoffmeister, F. Berger, M. Pogorzelskiy, G. Zhang, C. Zwick, and C. Müller-Birn. Toward cyber-physical research practice based on mixed reality. In M. Burghardt, R. Wimmer, C. Wolff, and C. Womser-Hacker, editors, *Mensch und Computer 2017 – Workshopband*, Regensburg, 2017. Gesellschaft für Informatik e.V.
- [22] J. Huang, R. White, and G. Buscher. User see, user point: gaze and cursor alignment in web search. In *Proc. of the SIGCHI Conference*, pages 1341–1350. ACM, 2012.
- [23] N. Ide and L. Romary. International standard for a linguistic annotation framework. *Nat. Lang. Eng.*, 10 (3–4): 211–225, Sept. 2004. ISSN 1351-3249. URL <http://dx.doi.org/10.1017/S135132490400350X>.
- [24] A. Jameson. Usability and the Semantic Web. In *ESWC*, pages 3. Springer, 2006.
- [25] N. Karam, C. Müller-Birn, M. Gleisberg, D. Fichtmüller, R. Tolsdorf, and A. Güntsch. A Terminology Service Supporting Semantic Annotation, Integration, Discovery and Analysis of Interdisciplinary Research Data. *Datenbank-Spektrum*, 16 (3): 195–205, Nov. 2016. ISSN 1610-1995. URL <https://doi.org/10.1007/s13222-016-0231-8>.
- [26] D. R. Karger. The semantic web and end users: What’s wrong and how to fix it. *IEEE Internet Computing*, 18 (6): 64–70, 2014.
- [27] A. Khalili, S. Auer, and D. Hladky. The rdfa content editor – from wysiwyg to wysiwym. In *Proceedings of COMPSAC 2012 – Trustworthy Software Systems for the Digital Society, July 16–20, 2012, Izmir, Turkey*, 2012. URL http://svn.aksw.org/papers/2012/COMPSAC_RDFaCE/public.pdf.

- [28] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2 (1): 49–79, 2004. ISSN 1570-8268. <https://doi.org/10.1016/j.websem.2004.07.005>. URL <http://www.sciencedirect.com/science/article/pii/S1570826804000162>.
- [29] S. Lohmann, S. Negru, F. Haag, and T. Ertl. Visualizing ontologies with vowl. *Semantic Web*, 7 (4): 399–419, 2016.
- [30] V. Lopez, M. Fernández, E. Motta, and N. Stierler. PowerAqua: Supporting users in querying and exploring the Semantic Web. *Semantic Web*, 3 (3): 249–265, 2012.
- [31] M. Marcińczuk, J. Kocoń, and B. Broda. Inforex – a web-based tool for text corpus management and semantic annotation. In N. C. C. Chair, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.
- [32] M. J. McGuffin and R. Balakrishnan. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *TOCHI*, 12 (4): 388–422, 2005.
- [33] E. T. Meyer and R. Schroeder. The world wide web of research and access to knowledge. *Knowledge Management Research & Practice*, 7 (3): 218–233, Sept. 2009. ISSN 1477-8246. URL <https://doi.org/10.1057/kmrp.2009.13>.
- [34] C. Müller-Birn, T. Klüwer, A. Breitenfeld, A. Schlegel, and L. Benedix. Neonion: Combining human and machine intelligence. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing, CSCW'15 Companion*, pages 223–226, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2946-0. URL <http://doi.acm.org/10.1145/2685553.2699012>.
- [35] J. L. Navarro-Galindo and J. Samos. The FLERSA tool: adding semantics to a web content management system. *IJWIS*, 8 (1): 73–126, 2012. URL <http://dx.doi.org/10.1108/17440081211222609>.
- [36] M. Neves and U. Leser. A survey on annotation tools for the biomedical literature. *Briefings in Bioinformatics*, 2012.
- [37] J. Nielsen. 10 usability heuristics for user interface design. Fremont: Nielsen Norman Group. [Consult. 20 maio 2014]. Disponível na Internet, 1995.
- [38] K. Ohara, T. Berners-Lee, W. Hall, and N. Shadbolt. Use of the semantic web in e-research. *World Wide Research: Reshaping the Sciences and Humanities*, page 130, 2010.
- [39] D. Oldman, M. Doerr, and S. Gradmann. Zen and the art of linked data. In R. S. a. U. Susan Schreibman, editor, *A New Companion to Digital Humanities*, pages 251–273. John Wiley & Sons, Ltd, 2015. doi:10.1002/9781118680605.ch18.
- [40] J. Pratt, P. V. Radulescu, R. M. Guo, and R. A. Abrams. It's alive! animate motion captures visual attention. *Psychological Science*, 21 (11): 1724–1730, 2010.
- [41] C. Preist. A conceptual architecture for semantic web services. In *ISWC*, pages 395–409. Springer, 2004.
- [42] G. Rizzo and R. Troncy. Nerd: Evaluating named entity recognition tools in the web of data. 2011.
- [43] A. Sears. Heuristic walkthroughs: Finding the problems without the noise. *Int. J. Hum. Comput. Interaction*, 9 (3): 213–234, 1997.
- [44] L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376. Springer, 2005.
- [45] H. N. Talantikite, D. Aissani, and N. Boudjlida. Semantic annotations for web services discovery and composition. *Comput. Stand. Interfaces*, 31 (6): 1108–1117, Nov. 2009. ISSN 0920-5489.
- [46] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semant.*, 4 (1): 14–28, Jan. 2006. ISSN 1570-8268.
- [47] K. van Deemter and R. Kibble. What is coreference, and what should coreference annotation be? In *Proceedings of the Workshop on Coreference and Its Applications, CorefApp '99*, pages 90–96, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. URL <http://dl.acm.org/citation.cfm?id=1608810.1608828>.
- [48] D. Vrandečić and M. Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 2014.
- [49] A. Widlöcher and Y. Mathet. The glozz platform: A corpus annotation and mining tool. In *Proc. of the DocEng*, pages 171–180. ACM, 2012.
- [50] M. Wilkinson and et al. The fair guiding principles for scientific data management and stewardship. *Nature Scientific Data*, (160018), 2016. URL <http://www.nature.com/articles/sdata201618>.

Bionotes



Andre Breitenfeld

Freie Universität Berlin, Human-Centered Computing, Berlin, Germany
andre.breitenfeld@gmail.com

Andre Breitenfeld has studied Computer Science at the Freie Universität Berlin. He is interested in interaction and service design. His current work focuses on digitalization and mobility services in the automotive industry.



Florian Berger

Freie Universität Berlin, Human-Centered Computing, Berlin, Germany
florian.berger@fu-berlin.de

Florian Berger is a research associate at the HCC group at the Freie Universität Berlin. His research is at the Cluster of Excellence “Image Knowledge Gestaltung – An Interdisciplinary Laboratory” at the Humboldt Universität zu Berlin, developing a mixed-reality object annotation system.



Ming-Tung Hong
Freie Universität Berlin, Human-Centered
Computing, Berlin, Germany
mingtung.hong@fu-berlin.de

Ming-Tung Hong is a PhD student in Computer Science at the HCC group at Freie Universität Berlin. Her research focuses on augmenting humans' sensemaking processes based on semantic annotations by allowing humans to interact with machine-recommended information.



Claudia Müller-Birn
Freie Universität Berlin, Human-Centered
Computing, Berlin, Germany
clmb@inf.fu-berlin.de

Claudia Müller-Birn is the head of the research group Human-Centered Computing (HCC) at the Institute of Computer Science at the Freie Universität Berlin. Her research is in the fields of Computer-Supported Cooperative Work, Human-Machine Collaboration, and Social Semantic Computing.



Maximilian Mackeprang
Freie Universität Berlin, Human-Centered
Computing, Berlin, Germany
maximilian.mackeprang@fu-berlin.de

Maximilian Mackeprang has recently completed his M.Sc. in Computer Science at the Humboldt Universität zu Berlin and is now member of the HCC group at the Freie Universität Berlin. His current research interests include Human-Centered Design, Creativity Support Software and Collaborative Ideation Systems.