

FREIE UNIVERSITÄT BERLIN

A State-of-the-Art Review of Semantic
Annotation Tools

Andre Breitenfeld, Claudia Müller-Birn

TR-B-17-3
March 2017



FACHBEREICH MATHEMATIK UND INFORMATIK
SERIE B • INFORMATIK

Abstract

The Web of Data is the vision about a worldwide knowledge base. A distributed knowledge base consisting of structured data that states facts about the world. The atoms of these facts are called entities. Entities can be related in almost any way to state facts. The resulting structure is a network from a large number of relationships. However, facts are not written in stone and may change. The network is in a constant state of change due to the exploration and publication of new knowledge sources. These knowledge sources, however, need to be pre-processed to become part of the Web of Data. Preprocessing comprises common activities like the identification of entities and the extraction of relationships between them. A technological cornerstone towards achieving this vision is the semantic annotation. Its goal is to make knowledge explicit by restructuring unstructured data with semantic information. Concerning the semantic preprocessing, the focus of this work is on the activity of relation extraction. We compare a number of tools and compare their interaction design for annotating relationships.

1 Introduction

Annotation is ubiquitous in a wide range of applications. The class of tools capable of creating and managing digital annotations is heterogeneous. It reaches from general purpose tools to sole purpose tools, predominantly for linguistic annotation¹ [2]. A classification of these tools is possible by different criteria, for instance by the degree of automation and the type of media a tool supports. An important distinction to make is whether a tool supports semantic annotation or not. Structured semantic annotation remains a special case of annotation above unstructured annotation like commenting, highlighting and tagging. The purpose of this section is to identify semantic annotation tools and to select those which are related to the research.

We focus in this study on relationship annotation, thus, we start with introducing this concept. Subsequently, an exposition of criteria will be presented as a decision-model to determine when a tool should be taken into further consideration, and then, a set of related tools will be presented concerning the selection criteria.

1.1 Annotation of Relationships

The notion of *relationship* is a universal concept to describe how arbitrary entities are linked to each other. Relational facts can be found in any contexts like images and texts. Words, sentences and paragraphs of a text can be related as well as contours, colors and motifs of an image. The application of relationships, in this work, focuses on relations embedded in written language. In this context, the annotation is the connection between text and fact. A single annotation represents the information carrier of a fact and it's textual anchors. Moreover, an annotation is not only information it is also a process with a defined workflow [19]. Casting a workflow to create relationships into an interaction design requires an a priori understanding of the activities necessary. Although the creation of relationships is a recurring task in semantic annotation there is no standardized workflow. A screening of Google Scholar provided no results on a general process. However, a process understanding can be derived from the perspective of relation extraction as IE task and from a reading perspective. The latter perspective has a special role since human-computer interaction and reading have one thing in common: both involve cognitive processes.

While reading a sentence a human being builds an understanding of it in two ways. The first way considers the examination of the structure of a sentence which atomically consists of words and punctuation. A sentence is being analyzed and divided into meaningful chunks according to its atoms. In this way, a complex sentence will be reduced into simpler parts. Besides that words are being classified, categorized and conceptualized. Eventually, the analysis is followed by a synthesis. The synthesis enables a reader to recognize how the parts in a sentence are grammatically related to each other. The second way of understanding considers the meaning of a sentence as whole: *What does it say and what does it mean*. Kurland [16, pp. 19 sq.] denotes both ways as the “two aspects of sentence awareness”. The dual awareness enables a reader to talk about what a text says and how the text does it.

¹Linguistic annotation concerns the enrichment of corpora with linguistic information, e.g. part-of-speech tagging [13]

In addition, Kurland [16, pp. 70 sqq.] shows different ways in responding to a text. He differentiates three modes of reading and discussing:

Restatement—a reader restates the original information of a text in own words. In this way, a reader proves the understanding of what is said.

Description—a reader describes the role of a text within a whole. In this way, a reader describes how sentences and paragraphs are interrelated.

Interpretation—a reader infers an overall meaning. In this way, a reader detects a meaning beyond what is said.



Figure 1: Explosion on the Deepwater Horizon²

Depending on the reading mode, a text may reflect different understandings since each mode serves a certain purpose. For example, considering the events on the Deepwater Horizon as shown in Figure 1 and the following sentence:

There was an explosion on Deepwater Horizon.

The restatement of the sentence above is just that there was an explosion on the Deepwater Horizon. The appropriate description is that the sentence reports an incident on an oil platform. An interpretation could be that oil enters the sea causing an environmental disaster. Eventually, the essence of a sentence is described by its entities and relations. Due to the nature of triple-based knowledge, the mode restatement is most applicable for this work. In the example, the terms Deepwater Horizon and explosion are entities within the sentence and the occurrences of each entity is called an *entity mention*. Semantically the entity Deepwater Horizon refers to an oil platform and explosion to a kind of accident. Both entities are interrelated since the explosion took place on the oil platform.

From the perspective of IE, the inherent process to extract statements is similar since IE processing relies on linguistics analysis. Culotta et al. [6] states the extraction of relationships “[...] is the task of discovering semantic connections between entities. In text, this usually amounts to examining pairs of entities in a document and determining (from local language cues) whether a relation exists between them.” An IE algorithm proceeds in two phases [6]. At first, an algorithm tries to identify all entities within a document. This refers to the classification, categorization and conceptualization of words known as *named-entity-recognition* [26, p. 94]. Then, the type of relation will be predicted for each pair of entities.

²<http://www.latimes.com/nation/la-na-oil-spill-timeline-pictures-photogallery.html>

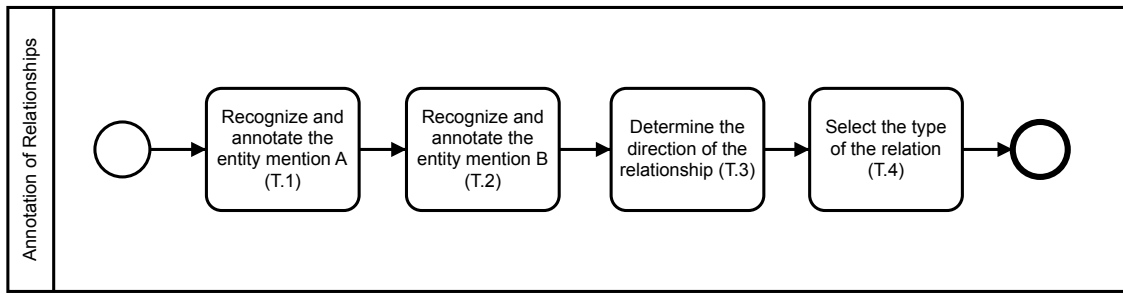


Figure 2: Process model for annotating relations with implicit relation mention

Transferring the steps described above to the area of semantic annotation involves a number of activities. The complete process model to create annotation-based relationships is depicted in Figure 2. In this process, the individual steps of recognition, conceptualization and annotation of entities are composed in activity **T.1** and **T.2**. The following activities form the *relation mention*. A relation mention describes how is the relation between two entities. In activity **T.3**, choosing the direction of the relationship corresponds to the order of the *subject-predicate-object* triple. The direction of a relationship between two entities constitutes the direction of the edge in the corresponding knowledge graph. The appropriate *predicate* will be chosen in activity **T.4**. Regarding the scope of this work, the activities **T.3** and **T.4** shall be facilitated by an interaction design.

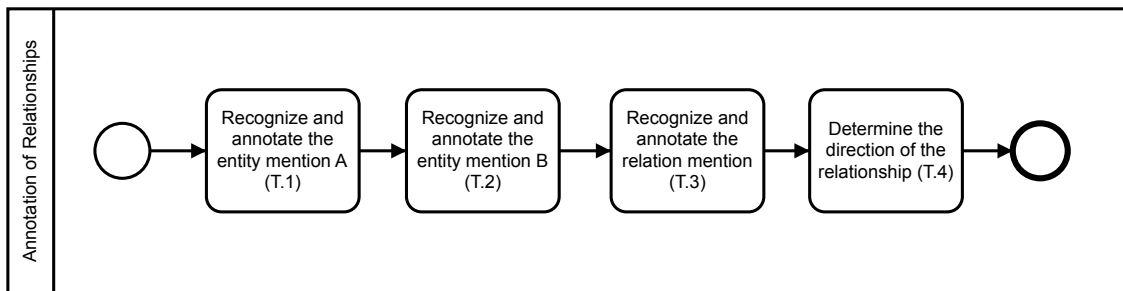


Figure 3: Process model for annotating relations with explicit relation mention

Alternatively, the general process can be extended to include the *explicit relation mention* as shown in Figure 3. An explicit relation mention is an explicit reference in the text stating the relationship between the entities [18, p. 3]. Therefore, activity **T.3** shall be substituted by an additional annotation step. This approach is useful to increase the granularity of a statement as each component of the relationship is referenced by a textual anchor. Additional granularity might improve the traceability of a fact for a higher trust in a statement. However, the process requires additional effort and is not applicable if there is no explicit relation mention in the text. This is the case when a relation is implicit in the text and a reader has to infer the relationship from contextual information. The subsequent state-of-the-art review will consider both options to annotate relationships. The adopted process model for this work will be discussed later on.

1.2 Criteria for Tool Selection

Due to the diverse tool landscape, the set of annotation tools needs to be limited by relevance. For the sake of this work, tools which are capable for creating semantic relations are relevant. Thus, the existence of an annotation scheme for relationships based on a controlled vocabulary is an essential criterion (Criterion **C.1**).

At this point, the degree of automation is important which can be distinguished according to manual and automatic annotation. Tools can be categorized according to the featured degree of automation. Basically, the selected degree depends primarily on the size of the corpora. Furthermore, it is a trade-off between the initial effort necessary to configure an automation workflow and

the effort to annotate the entire corpora at hand manually. Manual annotation is a time-consuming task and requires the effort of domain experts [24]. However, there is no out-of-the-box automation of corpus annotation and processing for all types of tasks. Human intelligence is indispensable when it comes to the deep interpretation of contextual information [1, p. 64]. Out of this reason and with respect to the goal of this work, the second criterion for a related tool requires a workflow for manual creation of relationships (Criterion **C.2**).

Next, this work focuses on relationships embedded in written language.

Due to the time constraint of this work, annotation tools for image, audio, video and 3D object annotation are excluded. Basically, the interaction designs in previously excluded tools are just as interesting. However, the corresponding tools will be not considered due to the time constraint of this thesis. Therefore, the next criterion requires the annotation of textual sources like plain or formatted text including web pages (Criterion **C.3**).

For the purpose of this work, it is irrelevant if a tool is still maintained by their developers. However, testing an interaction design based on screenshots is only possible to a limited extent – interactive aspects need to be tested hands-on. In order to review a tool and the interaction to create relationships it is indispensable to use the tool. Therefore, the last criterion constitutes the possibility to test a tool either by installing it from source code or using an online or binary distribution (Criterion **C.4**).

Code	Criteria
C.1	Supports semantic relationships from controlled vocabulary
C.2	Supports manual creation of relationships
C.3	Supports relation extraction from textual sources
C.4	Source code, online or binary distribution is available

Table 1: Criteria for tool selection

Table 1 summarizes the aforementioned criteria. A tool suitable for an in-depth review should meet all the criteria without exception.

1.3 Tool Selection

The primary sources for a selection of tools are research papers on semantic annotation tools found on Google Scholar and ACM Digital Library. Each tool found was checked individually to its suitability. The formal verification whether a tool satisfies a certain criterion was ensured by studying the user’s manual, feature description and web page of a tool.

The initial set of tools were screened from the doctoral thesis of [1]. Based on this work eleven tools for structured annotation were identified: Brat, Dexter, Glozz, Analec, CorpusTool, CATMA, GATE, Knowtator, MMAX2, WordFreak and WebAnno. After applying each criterion from Table 1 seven tools are considered to be relevant: Brat, Glozz, Analec, GATE, Knowtator, MMAX2 and WebAnno. Tools which do not meet all criteria are Dexter, CorpusTool, CATMA and WordFreak. Dexter, CorpusTool and CATMA do not support relationships neither for co-referencing nor in any other manner (**C.1** and **C.2** unsatisfied). WordFreak uses a plugin architecture to define custom annotation schemes including the annotation of co-references. However, it is unclear whether the tool supports the manual annotation of relations (**C.2** unsatisfied).

An additional set of tools were screened from the survey on annotation tools of [24]. In contrast to the work of [1], the authors focused on general purpose tools for manual annotation of semantic facts in biomedical literature. Semantic facts involve the annotation of species and chemical entities from biomedical research. Related tools should at least support the annotation of named entities and relationships optionally. The application of relationships between entities relates to the annotation of protein-protein interactions³ (PPIs) described in text. Based on the survey paper eight additional tools were identified: @Note, Argo, Bionotate, Callisto, Djangology, MyMiner, Semantator and XConc Suite. After applying the criteria, two additional tools will be considered in the review phase: Bionotate and XConc Suite. Tools which do not meet all criteria are Callisto,

³A physical contact between proteins caused by biochemical events [27]

Semantator, Argo, Djangology, @Note and MyMiner. Although Callisto and Semantator assert the support of relationship, the authors were unable to create relationships with the tools while building their test scenario (C.2 unsatisfied). Also, Semantator is closed source and a download of the tool was not possible due to a broken link (C.4 unsatisfied). Argo, Djangology and @Note do not support the creation of relationships (C.1 and C.2 unsatisfied). MyMiner supports binary relations by means of an interaction matrix. This kind of relation is used to indicate whether there could be an PPI, however, without any further semantic. An unspecified relationship will not be considered as a semantic relation (C.1 unsatisfied).

Literature research on Google Scholar yielded six additional semantic annotation tools: FLERSA [23], DOME0 [4], Pundit [9], RDFaCE [14], Inforex [20] and Vogon [8]. Tools fulfilling all criteria are Pundit, Vogon and Inforex. DOME0 and RDFaCE do not support the annotation of relationships (C.1 and C.2 unsatisfied). The research paper of FLERSA provides the evidence that the tool supports the creation of relationships. However, there is no source code and distribution available online. In this case, a hands-on test of the interaction design is not possible (C.4 unsatisfied).

Tool	Web	C.1	C.2	C.3	C.4	Source
@Note	https://omictools.com/note-tool	×	×	✓	✓	[24]
Analec	http://lattice.cnrs.fr/Analec,68	✓	✓	✓	✓	[1]
Argo	http://argo.nactem.ac.uk	×	×	✓	✓	[24]
Bionotate	http://bionotate.sourceforge.net	✓	✓	✓	✓	[24]
Brat	http://brat.nlplab.org	✓	✓	✓	✓	[1]
Callisto	http://mitre.github.io/callisto	×	×	✓	✓	[24]
CATMA	http://catma.de	×	×	✓	✓	[1]
CorpusTool	http://corpustool.com	×	×	✓	✓	[1]
Djangology	http://djangology.sourceforge.net	×	×	✓	✓	[24]
DOME0	http://annotationframework.org	×	×	✓	✓	[4]
FLERSA	-	✓	✓	✓	×	[23]
GATE	https://gate.ac.uk	✓	✓	✓	✓	[1]
Glozz	http://glozz.free.fr	✓	✓	✓	✓	[1]
Inforex	http://inforex.clarin-pl.eu	✓	✓	✓	✓	[20]
Knowtator	http://knowtator.sourceforge.net	✓	✓	✓	✓	[1]
MMAX2	http://mmax2.sourceforge.net	✓	✓	✓	✓	[1]
MyMiner	http://myminer.armi.monash.edu.au	×	×	✓	✓	[24]
Pundit	http://thepund.it	✓	✓	✓	✓	[9]
RDFaCE	http://rdface.aksw.org	×	×	✓	✓	[14]
Semantator	http://informatics.mayo.edu/cntro	×	×	✓	×	[24]
Vogon	http://gobtan.sourceforge.net	✓	✓	✓	✓	[8]
WebAnno	http://webanno.github.io	✓	✓	✓	✓	[1]
WordFreak	http://wordfreak.sourceforge.net	✓	×	✓	✓	[1]
Xconc Suite	http://geniaproject.org	✓	✓	✓	✓	[24]

Table 2: Evaluated tools from literature review

Eventually, twelve tools were selected with regard to the criteria for an in-depth review. Table 2 summarizes the set of tools.

The considered tools will be reviewed by means of defined attributes.

2 State-of-the-Art Review

In Section 1.3 it was shown that approximately half of the annotation tools support relationships whereas all tools support the annotation of named-entities. It can be assumed that the support of relationships is not uncommon but not always part of a minimal viable semantic annotation tool. Another insight is that the interaction to create relations is not a standardized process such as word processing where a cursor denotes the editing position. Different approaches exist in the tool

landscape. This section should provide an impetus for new ideas by a detailed investigation of the existing approaches.

This investigation was possible in two ways: firstly, concerning the usability according to [1]; secondly, concerning the solution concept according to [24]. Assuming the first way, an investigation on usability would reveal usability problems for each tool. Looking back on the work of [1], it had to be taken into consideration that a problem identified can be very specific for tool. The comparability between tools would be limited to the number of problems. Consequently, it was unclear how to innovate from a set of heterogeneous problems. In addition, a structured usability evaluation of all tools would exhaust the scope of this thesis. For this reason, the question how a tool does it was more important as the question how well a tool does it. Therefore, the investigation focused on the solution concept.

When it comes to the generation of ideas, techniques for ideation are often used. An analytical approach for the investigation of an existing solution represents the method *attribute listing* [12]. [5] developed the technique for designing and re-designing a solution. The idea of the technique is to decompose an existing solution into attributes. Attributes are features or properties of a solution and represent an unbiased feedback. The design of alternative solutions is accomplished by modifying individual attributes. In this context, attribute listing constituted the methodological basis of the review. The review proceeded in two steps:

1. Classification of the interaction design by interaction pattern
2. Characterization of research-related attributes

The first step represented a review on a higher-level. Focus here was on the mapping of the interaction to an interaction pattern. Interaction patterns are UI design guides similar to design patterns of software engineering. They offer a collection of best practices for recurring problems in the area of interaction design. The motivation of this step was to determine whether patterns are used and to identify common practices. As an inspirational source of patterns, the collection from [29] and [11] were consulted. In the second step, tools were reviewed on a lower-level related to the research topic. This step involved the attribute-based approach. The attribute-based review is not equal to an evaluation for assessing the quality. It is a descriptive approach in order to qualify and quantify specific traits and behaviors of a tool. In contrast to [24] who enumerated attributes on a general tool level, attributes were specific for the annotation of relationships. The used attributes were shared across the tools in order to ensure the comparability of the approaches.

In the following, the used attributes will be defined. Subsequently, tools and findings of the review will be presented.

2.1 Defining the Attributes

So far, the diversity of approaches were mentioned without describing the differences. The role of the attributes was to expose the range of diversity. Furthermore, the list of attributes represented the fundamental base to derive requirements for the new solution. For further use of these attributes, it was important to concentrate the review on the right aspects. With respect to the goal of this work, attributes were designed to reflect the quality goals simplicity and capability. The proposed set of attributes was not meant to be exhaustive but sufficient for the research. Each attribute represented a variable with a discrete value from a defined scale. A dichotomous scale⁴ was used to describe whether a certain feature is present in one way or another. Binary scales were used to decide whether a feature exists in general. This section provides an enumeration of attributes categorized into three groups: *feature-specific*, *interface-specific* and *interaction-specific* attributes.

The first group of *feature-specific* attributes concerned the capability of a tool. Critical features for capability were worked out from the annotation processes formalized in Section 1.1. The introduced processes include the entity mentions of the subject and object. Regarding this point, it had to be checked whether a tool involves the entity mentions (Attribute **A.1**). The process depicted in Figure 3 additionally involves the explicit relation mention. Additionally, it had to

⁴Nominal scale with two levels

be checked whether the annotation includes the explicit relation mention (Attribute **A.2**). The Web of Data is a collaboratively curated knowledge base. Provenance information is important for the trustworthiness of the data [10]. In this regard, it had to be checked whether the annotation includes any provenance information like creator and annotation date (Attribute **A.3**). The last attribute in this category relates to the origin of the terminological terms. An important idea of the Semantic Web proposes the integration of external ontologies. A common terminology is crucial for a knowledge base integrated in the Web of Data rather than isolated. In the context of semantic annotation, relations are associated with the corresponding resource of the external relationship. It had to be checked whether the tool incorporates relationships from external ontologies (Attribute **A.4**).

Feature-specific attributes		
Code	Attribute	Scale
A.1	Relationship annotation includes entity mention	Yes/No
A.2	Relationship annotation includes explicit relation mention	Yes/No
A.3	Relationship annotation includes provenance information	Yes/No
A.4	Relationship from ontological properties	Yes/No

Table 3: Feature-specific attributes

The later two groups concerned attributes which might affect the simplicity of a solution. This mainly referred to the cognitive load imposed by the visual resources of the interface. The group of *interface-specific* attributes concerned structural and behavioral features of the interface for the creation. Structural features concerned the accessibility of the interface. At first, it had to be checked whether the interface offers a special mode to create relationships. Which means without accessing this mode it is not possible to create a relationship (Attribute **A.5**). In addition, the relationship between the interface to create a relation and text was another point. It had to be checked whether the interface is separated from the text or integrated into the text (Attribute **A.6**). Behavioral features related to the physical size and the location of the interface. The first aspect considered the way the interface allocates the space on the screen. Firstly, screen space can be allocated *constantly* independent from the entities. Secondly, screen space can be allocated *proportionally*, for example, in relation to the entities (Attribute **A.7**). The second aspect considered the spatial positioning of the interface. Firstly, the interface can be placed at a *fixed* position on the screen. Secondly, the interface can be placed *relative*, for example, relative to the entities in the text (Attribute **A.8**).

As the size of the terminological terms increases the combination of possible relations increases. The user has to memorize the possible relations for each pair of concepts. Any attempt to create an indefinite relation results in an error. A possibility to minimize the memorization represents a user assistance which visualizes the space of opportunities. The existence of a user assistance had to be checked in the tools (Attribute **A.9**). Also, abstraction is an important factor when it comes to the simplification of complex subjects such as relationships. As described prior to this chapter, a relationship is a triple of subject, predicate and object. This highly technical view on the data format does not entail any abstraction. Therefore, it had to be checked whether a tool uses the *subject-predicate-object* (SPO) paradigm in the interface directly (Attribute **A.10**). Consequently, this raised the question whether the interface provides an abstraction. It had to be checked whether a tool provides an abstraction layer for relations (Attribute **A.11**).

The last group of *interaction-specific* attributes concerned explicit and implicit properties of the interaction design. Explicit properties were features of the interaction design where implicit properties result from the interaction workflow. A first explicit property concerned to the role of the entity mentions when it comes to the creation of a relation. It had to be checked whether the interaction involves the visual anchors of the entities mention in a way (Attribute **A.12**). An implicit property lies in the effect of the interaction order on the direction of the relationship. It had to be checked whether the order of interactions implicitly determines the direction (Attribute **A.13**). Another implicit property represents the effort of the physical interaction with the mouse. On the one hand, this can be assessed by the minimum amount of clicks necessary to create a

Interface-specific attributes		
Code	Attribute	Scale
A.5	Interface offers a special mode for relations	Yes/No
A.6	Interface is separated from the text	Yes/No
A.7	Screen space allocation of the interface	Constant/Prop.
A.8	Spatial positioning of the interface	Fixed/Relative
A.9	Interface offers user assistance to avoid errors	Yes/No
A.10	Interface uses the subject-predicate-object paradigm	Yes/No
A.11	Interface uses a central metaphor	Yes/No

Table 4: Interface-specific attributes

relationship (Attribute **A.14**). And on the other hand regarding the movement of the cursor on the interface. However, operationalizing the cursor movement was difficult since factors like screen resolution and spatial position of the entities are varying.

Interaction-specific attributes		
Code	Attribute	Scale
A.12	Interaction involves the visual anchors	Yes/No
A.13	Interaction order effects the relation direction	Yes/No
A.14	Interaction effort in mouse clicks	Click count

Table 5: Interaction-specific attributes

The consideration of attributes from an feature-specific (see Table 3), interface-specific (see Table 4) and interaction-specific (see Table 5) perspective compiled a set of fourteen attributes.

2.2 Individual Review

In the following section, a tool review will be presented according to the attributes in Section 1.3. A total of twelve tools were reviewed individually as a mental foundation for the subsequent conception. In the following, the general process of annotating relationships are introduced, which provide the focus for the tool comparison.

2.2.1 Analec (Version 1.5)

Analec [17] is a sole purpose tool for linguistic annotation. The interaction pattern in Analec could be classified as *slot-filling*⁵ or *fill in the blanks*⁶ respectively. The user has to fill drop-down fields according to the SPO-paradigm. It allows the annotation of entity mentions as well as relation mentions by means of the feature *unit*. A unit can be an arbitrary item like an entity or relation. Each visual anchor gets a unique ID and can be selected from a drop-down field. The feature *schema* enables the creation of complex relations including the relation mention. Although the annotation scheme is customizable, it is unclear whether it allows relations from external ontologies due to the lack of documentation. Provenance information is not included since the tool provides no user authentication.

Relations are a special mode accessible via the main menu bar. Once it has been activated, the interface appears separated from the text at a fixed position. It has a constant size containing different drop-down fields. On the one hand, drop-down fields are used to filter entities and relations by type; on the other hand for selecting anchors and relations by ID. Figure 4 depicts a linguistic dependency between an expression (element 1) and a noun (element 2). The purpose of the other fields was unclear due to the lack of documentation and localization. Consequently, the interface

⁵<http://www.dfki.de/etai/SpecialIssues/Dia99/araki/araki/node3.html>

⁶<http://ui-patterns.com/patterns/FillInTheBlanks>

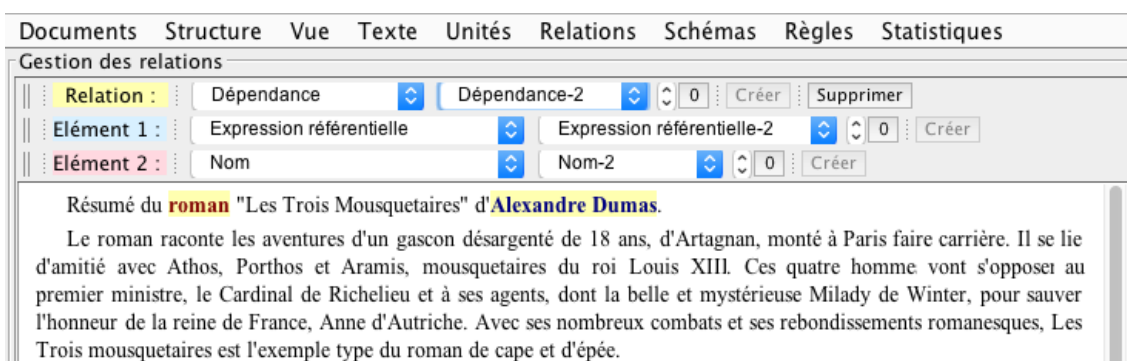


Figure 4: Interface of Analec

uses the SPO-paradigm with no further abstraction. User errors are prevented by the circumstance that it is possible to create relations without any constraints. Insofar there is no need for a user assistance. The user has to interact with each drop-down field. Alternatively, it is possible to click on the anchors to select 'element 1' and 'element 2'. The order of interaction has no effect on the direction since it is possible to modify the fields in any order. Regarding the effort in clicks, the user has to perform five clicks: two for selecting the elements, two for expanding and selecting the relation type and one for saving the relation.

2.2.2 BioNotate (Version 1.0)

BioNotate [3] is a sole purpose tool for the annotation of snippets from biomedical literature. The intended use of the tools is the annotation of relationships between biomedical entities. It allows the annotation of snippets extracted from biomedical literature where each snippet contains a potential fact. Relationships are realized by a *question-answer* interaction pattern. The user has to answer yes/no questions on each snippet of the form "does this snippet imply a direct interaction between the provided entities?". Questions are defined in advance by the research setup. Regarding the annotation, the process recommends the annotation of the entities and the relation in the text in order to support the traceability. Therefore, entity and relation mentions are included but optional. Ontological properties are not supported instead the existence and the type of relation is determined by a machine-unreadable answer. Provenance information is not included since the tool provides no user authentication.

Snippet

Extracted from article: PubMed [015632144](#)

Entities of interest:

GENE : Dab1

GENE : very low density lipoprotein receptor

Disabled-1 (Dab1) is an essential adaptor protein that functions in the Reelin signaling pathway and is required for the regulation of neuronal migration during embryonic development . Dab1 interacts with NPXY motifs in the cytoplasmic tails of the lipoprotein receptors ApoER2 and very low density lipoprotein receptor through an amino-terminal phosphotyrosine binding (PTB) domain (2).

Mark selected text as:

GENE INTERACTION

Does this snippet support a direct interaction between the provided entities?

- Yes, there is explicit positive evidence of interaction (please highlight support).
 No, there is explicit negative evidence of interaction (please highlight support).
 No, the two entities just co-occur.

Figure 5: Interface of BioNotate

In BioNotate the creation of relationships is not a special mode. It provides a split interface with the annotation snipped in the upper screen area and the interface to create the relationship in the lower screen area (see Figure 5). The area containing the answers is visible at any time at a fixed position on the bottom of the screen and appears at a size proportional to the numbers of answers. User errors caused by indefinite relations are prevented by design since possible relations are given as answer options. Regarding the abstraction, the SPO-paradigm is not present in the interface. Instead, BioNotate adds an abstraction layer by the question-answer pattern. Although the annotation of entity and relation mentions are recommended, visual anchors are not involved in the interaction to create the relation. The interaction order has no effect on the direction of the relationship since it is determined by the question and the given answer. The preparation of a relationship requires two clicks: one for choosing the answer and one for saving the annotation.

2.2.3 Brat (Version 1.3)

Brat [28] is general purpose annotation tool. A special feature of the tool is the visualization of the textual fragments which splits the entire text by sentence and arrange these vertically. Relationships are realized by a *drag-and-drop*⁷ interaction pattern. The user has to connect two entities by means of drag-and-drop gesture. In Brat the annotation includes the corresponding entity mentions. However, the annotation of relation mentions is not supported. Furthermore, ontological properties are not supported since there is no possibility to associate relations with external properties. The tool provides a user authentication, however, it is unclear whether the annotation includes provenance information.

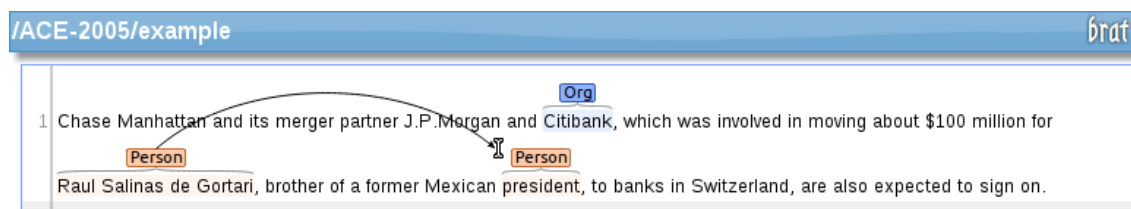


Figure 6: Interface of Brat⁸

Relationships are not a special mode but accessible by start dragging an entity. While dragging, a connecting line appears following the position of the cursor (see Figure 6). The connecting line is considered to be a metaphor for relations. The second step is to release the mouse while the cursor is above a second entity. On completion of the second step, a dialog appears at a fixed position in the center of the screen at a constant size. The dialog represents the final point to complete the process. In this regard, the interface for creating the relation is separated from the text. Essentially, the content of the dialog is based on the SPO-paradigm showing the subject, object and predicate of the relation (see Figure 7). In this final step, the user has to choose the type of relation and submit the dialog. There is no visual feedback in case there is no common relationship between the entities. Whenever the user tries to create an indefinite relation, the connecting line disappears without showing the dialog.

The order of interaction determines the direction of the relation where the dragged entity becomes the subject and the dropped entity the object. Once the drag-and-drop operation completes, it is not possible to change the direction in the dialog (see Figure 7). The interaction involves the visual anchors necessarily since the drag-and-drop operation is an essential part of the interaction design. Regarding the effort in clicks, the creation of a relation requires three clicks: one for the drag-and-drop operation, one to choose the type of relation and one to submit the dialog.

2.2.4 Gate (Version 8.2)

GATE Developer [7, pp. 37 sqq.] is a general purpose annotation tool. It is part of GATE family of products for natural language processing. The tool can be used for manual annotation and pre-

⁷<http://ui-patterns.com/patterns/drag-and-drop>

⁸<http://brat.nlplab.org/manual.html>

⁹<http://brat.nlplab.org/manual.html>

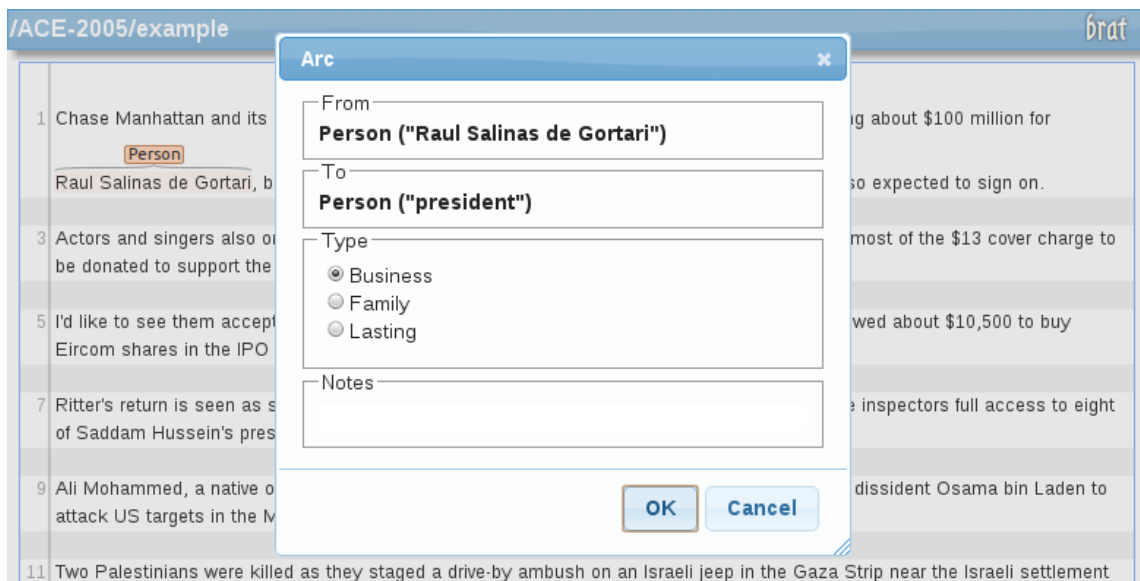
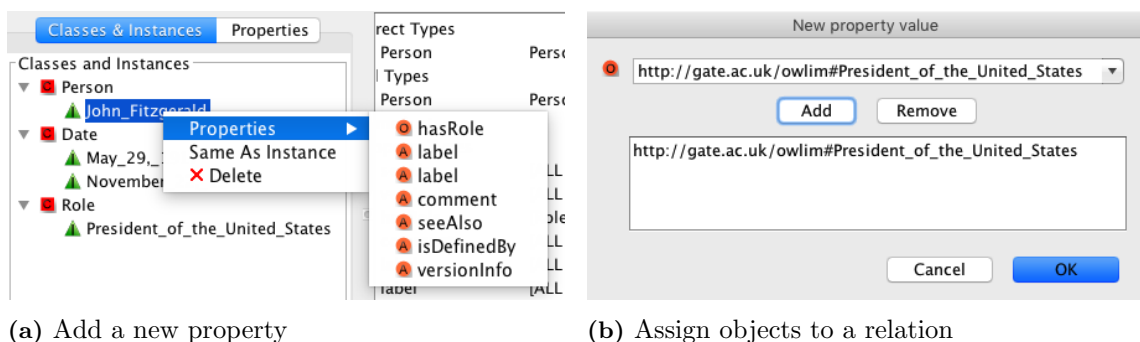


Figure 7: Relation dialog in Brat⁹

annotation of rules for automatic annotation and relation extraction. In addition it provides a build-in ontology editor which enables the user to create and manage classes, instances and properties. Furthermore, it is also possible to integrate external ontologies. Regarding the annotation, GATE allows the annotation of entity mentions but no relation mentions. Entity mentions are used to populate entities in the underlying ontology. However, entity mentions are not included in the relation between the entities. Provenance information is not included since the tool provides no user authentication.



(a) Add a new property

(b) Assign objects to a relation

Figure 8: Create a relation in GATE

The interaction design in GATE does not correspond to a common interaction pattern. With regard to the annotator interface, relations were not visualized in the annotator interface. Instead, relations are a special mode integrated into the ontology editor of the tool. Insofar the interface is totally separated from the text. The interface to create a relation is accessible via the instance list by choosing a property from the context menu (see Figure 8a). After that, a dialog appears at a fixed position and a constant size. In this dialog, the user has to choose the object of the relation by adding one or more entities from the drop-down field (see Figure 8b). As shown in Figure 8b the interface neither uses the SPO-paradigm nor an abstraction by a metaphor. In case there are no suitable entities for the corresponding relation the dialog appears with an empty set of entities. Which means there is no user assistance preventing errors.

Visual anchors are not involved in the interaction due to the strict separation between the annotation interface (see Figure 9) and the ontology editor. Given that the ontology editor is opened, the preparation of a relation requires six clicks: one right-click on the instance in the

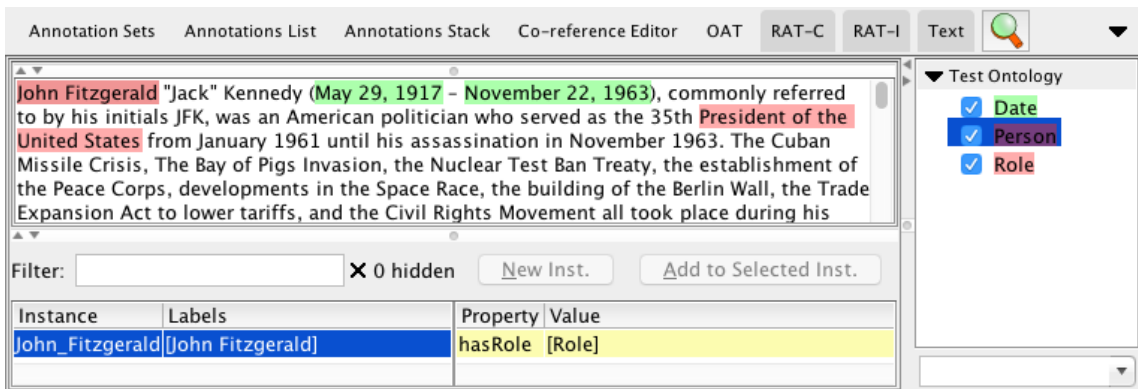


Figure 9: Interface of GATE

"Classes & Instances" list, one click on the desired property, two clicks in drop-down field to choose the instance, one click on "Add" and one click to submit the dialog.

2.2.5 Glozz (Version 2.1)

Glozz [31] is a general purpose annotation tool. The tool shares the same data model as Analec which consists of the concepts unit, relation and schema [21]. Insofar the annotation of entity and relation mentions is possible where both can be included in the relation. Another feature is that Glozz supports directed and undirected relations, however, with no difference in the creation process. Regarding the interaction, the interaction pattern is similar to Brat, however, it does not rely on drag-and-drop. The annotation scheme is customizable externally from file but there is no way to associate relations with ontological properties. Provenance information are included in the annotation.

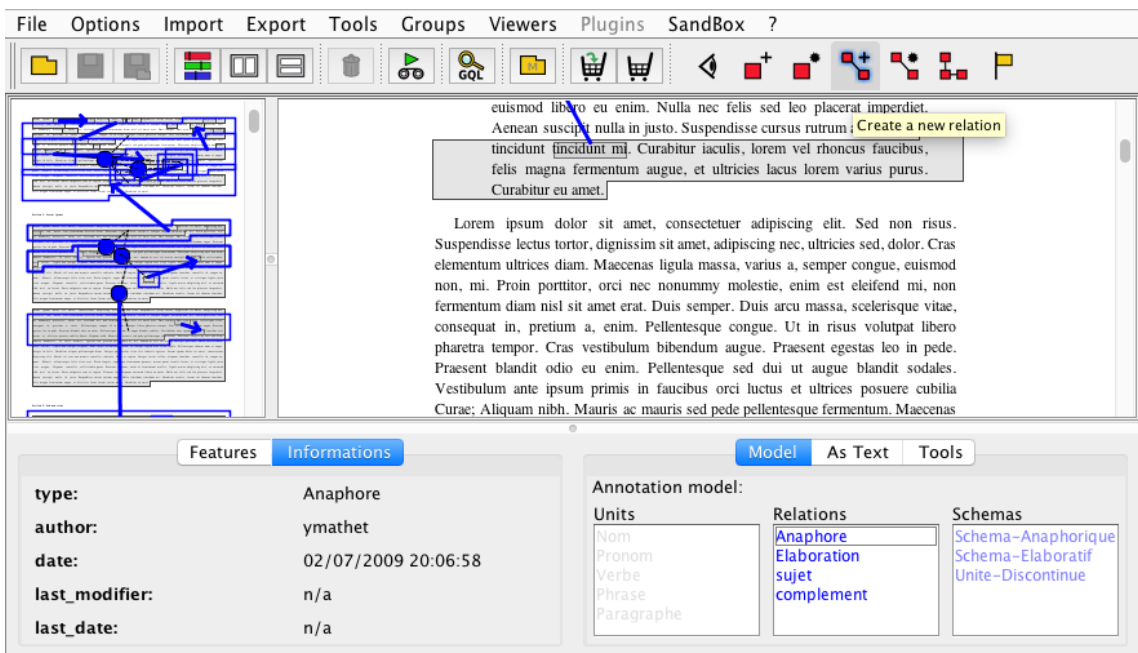


Figure 10: Interface of Glozz

Relations are a special mode accessible via the button "Create a new relation" in the toolbar (see Figure 10). In contrast to Brat, the selection of the relation type happens in advance. Once the relation type is selected (see Figure 10 in the lower screen) all relations will be created with

the selected type. As in Brat, Glozz uses the metaphor of the connecting line to visually connect entities. The connecting lines appear integrated into the text after selecting the source anchor (see Figure 11). While moving the mouse the line covers the space from the source anchor to the cursor. After selecting the target anchor the new relation will then be created. The SPO-paradigm is not present in the interface. User errors are prevented by design since the creation of relations is possible without any constraints.

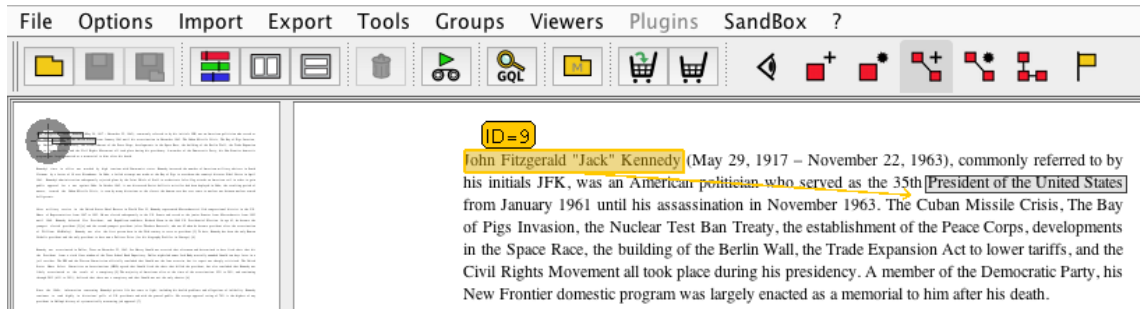


Figure 11: Create a relation in Glozz

The interaction design requires clicking on the visual anchors to draw the connecting line. As in Brat, the relation direction is determined from the moment the user selects the source anchor. Regarding the effort in clicks, the creation of a relation requires four clicks: one to enable the relation mode, one to choose the type of relation from a list, one to select the source anchor and one to select the target anchor.

2.2.6 Inforex (Reviewed at 09-20-2016)

Inforex [20] is a general purpose annotation tool. The tool offers a suite of different perspectives for a variety of purposes. Relevant for this work was the annotator perspective. It allows the annotation of entity mentions but no relation mentions. The tool has no common interaction pattern involved in the creation process. Instead, the interaction can be characterized as a preset click path on the entity mentions. Entity mentions are included in the relation. Regarding the involvement of ontologies, it is unclear whether relations are associated with external properties since the annotation scheme is predefined and not customizable. The tool provides a user authentication, however, it is unclear whether the annotation includes provenance information.

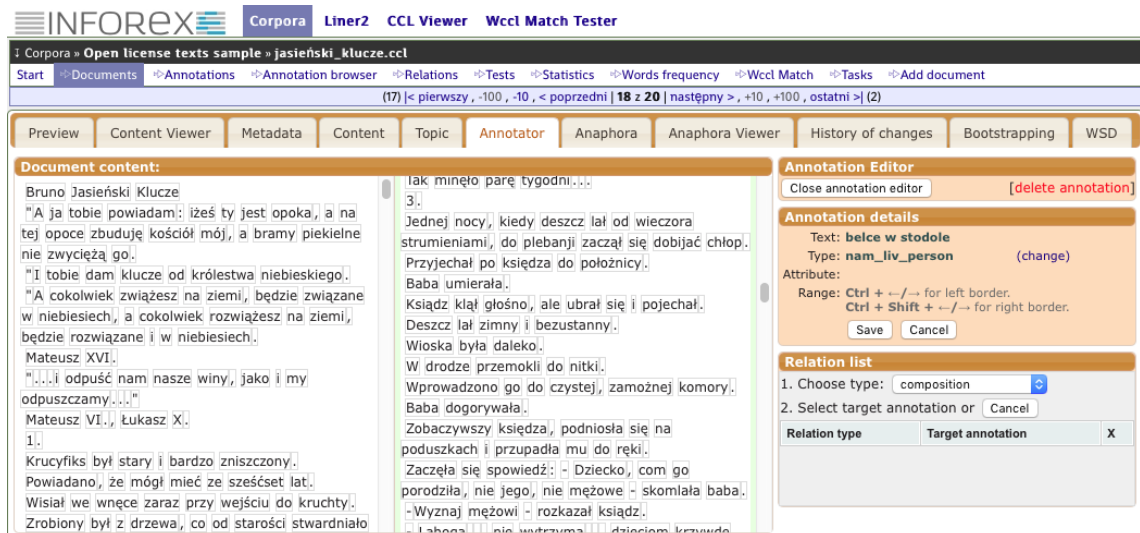


Figure 12: Interface of Inforex

Relationships are a special mode accessible by clicking an entity mention. Once an entity becomes selected, an annotation editor appears showing details on the annotation and a list of existing relations. This editor appears at a fixed position and constant size separated from the text. Within the list of relations a button named "Add Relation" starts the creation process (see Figure 23). Subsequently, the user has to select the type of relation. The process completes once the user selects the second entity mention (see Figure 12). In essence, the interface uses SPO-paradigm without further abstraction. The tool provides a user assistance. Visual anchors whose entities are not supported by the selected relation type become grayed out (see Figure 24). The interaction design involves the visual anchors necessarily and begins with selecting the subject of the relation. Insofar the interaction order determines the direction of the relation. Due to the interaction design, the preparation of a relation requires five clicks: one click to select the subject, one click on 'Add relation', two clicks for choosing the relation type from the drop-down field and one click to select the object.

2.2.7 Knowtator (Version 1.9)

Knowtator [25] is a plugin for Protégé which enables semantic annotation. The interaction pattern in Knowtator could be classified as *slot-filling*. Initially, the user creates a blank relation having a type but no subject and object (see Figure 22). After that, the relation gets completed by assigning the entity mentions of the subject and object. In addition it is possible to create complex relations including the relation mention. Knowtator benefits from Protégé in terms of the ontological infrastructure which is based on OWL¹⁰. Insofar, relations are associable with external properties. Provenance information is not included since the tool provides no user authentication.

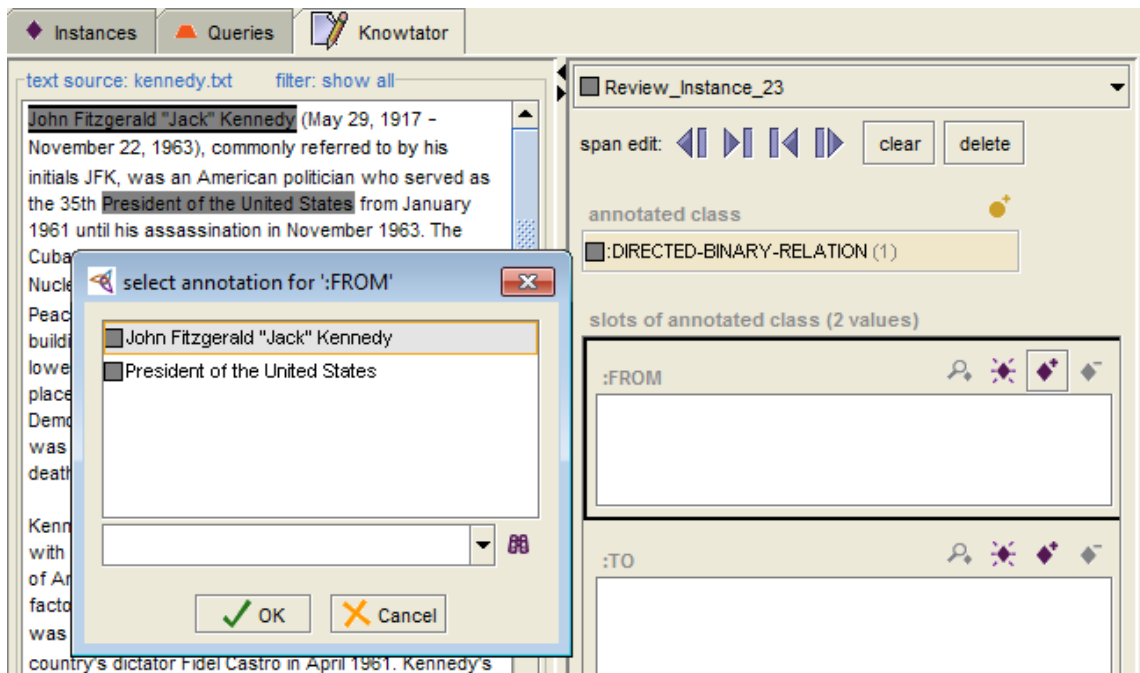


Figure 13: Interface of Knowtator

Relations are not a special mode in Knowtator. In general, the interface is always visible separated from the text at a fixed position on the right of the screen. It appears at a constant size and uses the SPO-paradigm (see Figure 13) with no further abstraction. Once the user selects a relation the slots of the subject (":FROM" field) and object (":TO" field) may be changed. Regarding the interaction, visual anchors are not involved in the interaction since the selection is made from a list. User errors are prevented by showing suitable entities exclusively in the selection dialog. In addition, the interaction order has no effect on the direction which can be changed at

¹⁰Web Ontology Language

any time. Due to the demanding interaction with the interface, ten clicks are required in total: two clicks to create a blank relation from the context menu, two clicks to select the relation from the drop-down field, three clicks each to assign the subject and the object.

2.2.8 MMAX2 (Version 1.13.003)

MMAX2 [22] is a general purpose tool for the annotation of XML documents. The identification of a common interaction pattern was not possible. According to Inforex, the creation process could be characterized as a fixed sequence of clicks. An entity mention is called *markable* in MMAX2. The markable is an essential part of the relation but the annotation of relation mentions is not supported. The creation proceeds in three steps. At first, the user has to select an entity by clicking on the markable with the left mouse button. In case the clicked markable overlaps with other markables, a selection menu appears showing the overlapping entities (see Figure 14). Now the first markable becomes focused. The user has to determine the second markable by using the right mouse button. After that, the user has to select the relation type from the appearing menu. A special feature is that the tool provides the relation types *markable-set* and *markable-pointer*. A relation defined as *markable-set* represents an undirected relation between entities where the *markable-pointer*, on the other hand, represents a directed relation. However, the definition does not allow the association with external properties. Provenance information is not included since the tool provides no user authentication.

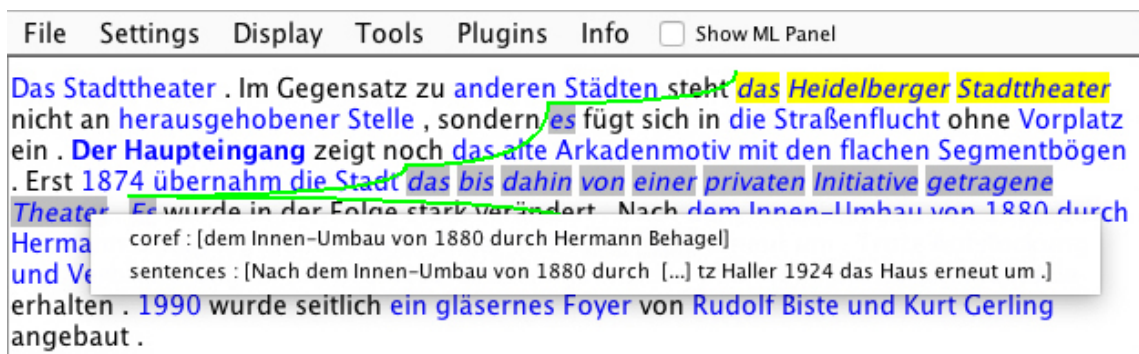


Figure 14: Interface of MMAX2

In MMAX2, the creation of relations is not a special mode. The interface appears above the text, more precisely relative to the second markable (see Figure 25). The size of the menu is proportional to the number of possible relations. If there are no common relations the interface will not appear. Insofar there is no user assistance. The SPO-paradigm is not present in the interface. As in Brat and Glozz, MMAX2 uses the metaphor of the connecting line (see Figure 14). In contrast to Brat the line is not visible while creating the relation but after completion. Regarding the interaction, visual anchors are involved in the interaction since the creation process requires clicking the markables. As described above, MMAX2 supports directed and undirected relations. Independent of the type, the process remains the same. In the former case, the interaction order determines the direction. By clicking the first entity the subject of the relation is determined immutably. Overall, the preparation of a relation requires three clicks according to the three steps.

2.2.9 Pundit (Version 0.0.45)

Pundit [9] is a general purpose tool for the annotation of web pages. A special feature of Pundit is the *Triple Composer*. The Triple Composer represents the central component to create statements of any kind. It allows the annotation of entities and creation of relational facts including the entity mentions. The annotation of relation mentions is not supported. Semantic annotation in Pundit provides relations from a predefined annotation scheme. However, the default annotation scheme is extensible by custom predicates which can be linked to properties from external ontologies. Provenance information is included in the annotation.

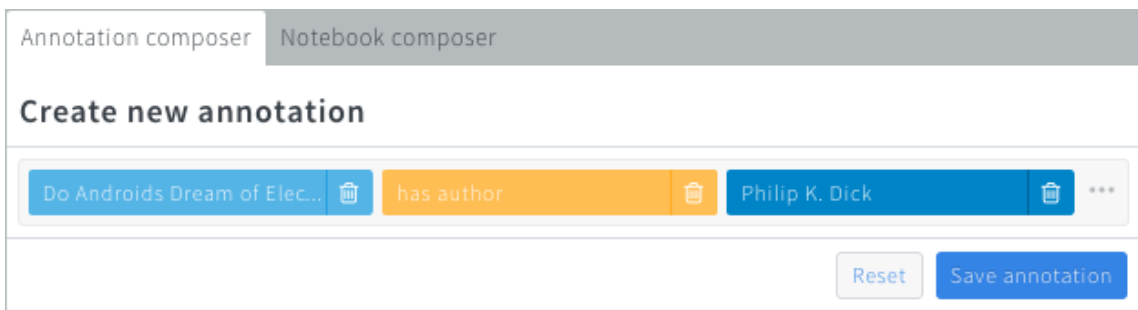


Figure 15: Triple composer in Pundit

The interface of the Triple Composer shown in Figure 15 contains three slots. The slots are intended for the subject, predicate and object of the relation. Insofar the interaction pattern in Pundit could be classified as *slot-filling*. The user has to fill the slots with values from the text or a search field. In this way, the Triple Composer serves a double purpose. On the one hand, it is possible to annotate named-entities and link these to external entities. And on the other hand, it is possible to create relations between these entities via the SPO-paradigm. Due to the double purpose of the Triple Composer, relations are not a special mode in the tool. The interface of the Triple Composer is separated from the text and allocates a constant screen space.

Do Androids Dream of Electric Sheep?

From Wikipedia, the free encyclopedia



Figure 16: Interface of Pundit

The visual anchors are necessarily part of the interaction in order to fill the slots of the composer. However, visual anchors are not reusable for filling the slots. The user has to re-annotate the text in order to select the subject and object according to the context menu in Figure 16. Since the user controls whether the selected text should be used as subject or as object (see Figure 16) the interaction order has no effect on the relation direction. While composing the relation Pundit lacks a metaphorical design. Instead, the Triple Composer uses the SPO-paradigm exclusively. There is no user assistance avoiding errors. When the user fills the slots for subject and object with entities not having a common relation the drop-down field for the predicate will not offer any values. Since existing visual anchors cannot be reused, the effort in clicks requires eight clicks: two clicks to highlight and choose the subject, two clicks to highlight and choose the object, three clicks in the search field to choose the predicate (includes an additional click on the "Use" button as shown in Figure 26) and one click to save the annotation.

2.2.10 Vogon (Version 1.0)

Vogon [8] is a general purpose annotation tool. The interaction pattern in Vogon could be classified as *slot-filling*. According to Pundit, the user has to complete slots for subject, predicate and object. In Vogon, entity and relation mentions are called *appellations*. Both types of appellations are included in the relation. A special feature are *baseless appellations* which enables relations without

textual mentions. In this way, it is possible to create statements which are not present in the text but inferred from the overall meaning. The tool has no build-in annotation scheme and requires the integration of external ontologies. Provenance information is included in the annotation.

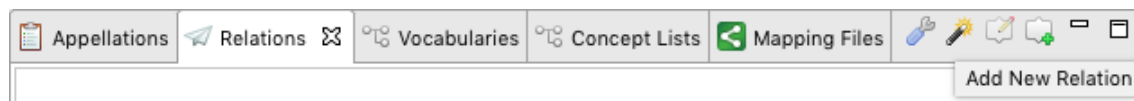


Figure 17: Relation tab in Vogon

Relations are a special mode accessible via the "Relations" tab. By clicking the button "Add New Relation" a dialog centered on the screen appears. The dialog provides a side by side separation of SPO-slots and the annotated text (see Figure 17). The drop-down slots can be filled with the appellations from the given text. In addition, it is possible to select the visual anchors to complete the slots. The interaction order has no effect on the direction since the slots can be changed at any time. After submitting the dialog the relation gets created.

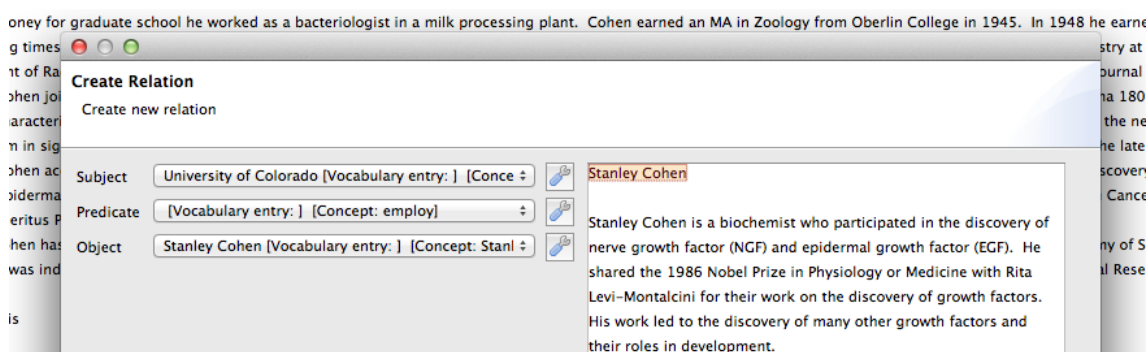


Figure 18: Dialog to create a relation in Vogon

Figure 18 provides the evidence that Vogon uses the SPO-paradigm. In contrast to Pundit, relations can be created without constraints which means any SPO-triple is valid. Insofar there is no need for a user assistance. Additionally, the tool offers a visual editor to create relations (see Figure 27). The visual editor is basically the place to create relations from baseless appellations. In this editor the metaphor of the connecting line is present. Regarding the effort in clicks using the "Create Relation" dialog, the preparation of a relation requires eight clicks: one click on "Add New Relation", two clicks each to choose the subject, predicate and object and one click to submit the dialog.

2.2.11 WebAnno (Version 2.3.1)

WebAnno [32] is an annotation tool for linguistic annotation but not exclusively. The tool is based technologically on Brat but extends it by features for collaboration. It introduces a new interface, however, the annotation interface is identical to Brat (see Figure 28). The same applies to the functionality for creating relationships. Only the dialog for specifying the relation type is slightly different but fulfills the same purpose (see Figure 29). Consequently, interaction pattern and values of the attributes are the same as for Brat. In contrast to Brat, WebAnno includes provenance information in the annotation.

2.2.12 XConc Suite (Version 1.0.22)

XConc Suite [15] is a sole purpose tool for the annotation of biomedical literature. The intended use of the tools is the annotation of biomedical entities such as proteins and triggers. In this context, proteins refer to named-entities and triggers to the remaining tokens in the text. A single trigger can give the evidence of a potential interaction between proteins. Each trigger found represents a biomedical event characterized by a *type*, *cause* and *theme*. The latter two fields represent

predefined relations to entities an event can have [30]. Provenance information is not included since the tool provides no user authentication.

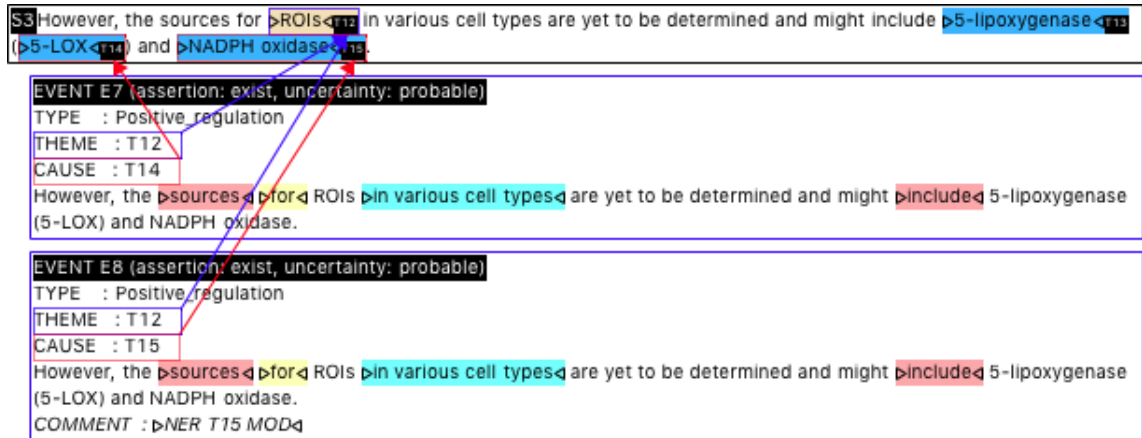


Figure 19: Interface of XConc Suite

While discovering biomedical events the task of the user is to connect each event field with the corresponding proteins and triggers in the text. Figure 19 shows a sentence (S3) containing two events (E7 and E8). Each event has a corresponding event block beyond the sentence. Both events are visually connected to the anchors in the text via the fields. Initially, the fields are empty and need to be filled by the user. Insofar the interaction pattern in XConc could be classified as *slot-filling*. Since each event corresponds to a trigger with a textual token, the trigger can be seen as the relation mention which is part of the whole event. The tool has no build-in annotation scheme and requires the import of external ontologies in OWL format.



Figure 20: Selection mode in XConc Suite

Creating a relation between a single slot and the text is a special mode accessible via the "Idref Selection Mode" in the toolbar (see Figure 20). Once the mode is enabled the selected slots can be connected by double clicking the visual anchors. The slots of an event block are the only visual components of the creation interface which are always visible at a constant size. Since the event blocks are positioned relative to the sentence the interface is considered to be integrated into the text. The SPO-paradigm is not present in the interface but a metaphor for relations. On completion, a connecting line appears indicating the connection between the slot and the text. Due to the fact that it is possible to connect any annotation with the slots theme and cause there is no need for a user assistance. The preparation of a single event relation requires four clicks: one click for selecting a slot, one click on the Idref Selector and a double click on the annotation.

2.3 Findings

This section reviewed a total of twelve annotation tools. As expected, the range of approaches was broad. Table 6 summarizes the results of the review by tool and attribute. The general process model in Figure 2 was the predominant model. All tools included the entity mention while the majority of tools does not include the explicit relation mention. Only four tools provided provenance information on the annotation.

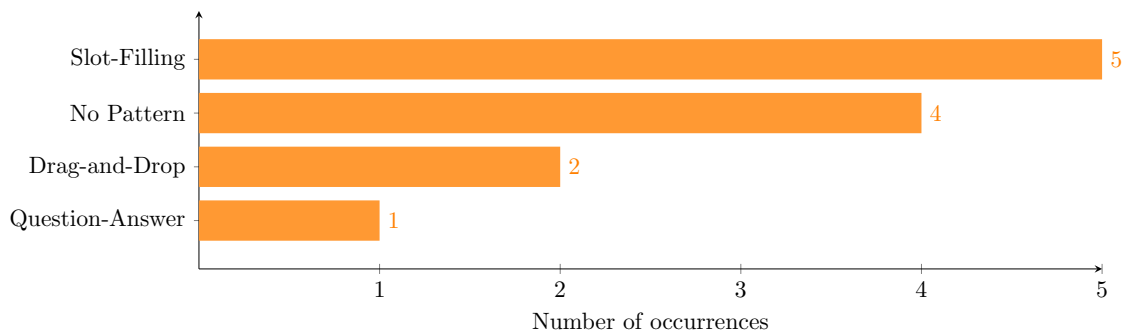


Figure 21: Number of occurrences per interaction patterns

Table 21 shows the number of occurrences per identified interaction pattern. Slot-filling was the most frequent interaction pattern followed by no common pattern. Drag-and-drop were identified in the tools Brat and WebAnno. The question-answer pattern identified in BioNotate was the simplest solution encountered.

In essence, the review showed two major groups: firstly, the group following the SPO-paradigm; secondly, the group using some kind of abstraction layer. However, the second group is very heterogeneous. Overall, the reviewed solutions lack a coherent design. Almost all tools provide a separation between text and the interface to create the relation (for example the Triple Composer in Pundit). The separated interface mainly appears at a fixed position on the screen. This creates the situation that the cursor is constantly moving between the interfaces in order to reach a specific feature. Consequently, the mouse movement is subjectively seen high in these tools. Regarding the click effort, the review pointed out cumbersome approaches like Knowtator, Vogon and Pundit. In addition, some tools have a screen-space inefficient visualization like Brat, WebAnno and XConc Suite.

Finally, the review detected two considerable design gaps towards a simple solution. The first design gap concerned an adequate abstraction level. Only four tools provide an abstraction layer while the majority of tools uses the SPO-paradigm or a different approach. The second design gap concerned the assistance of the user. User assistance is another point since only two tools provide precautions to prevent errors. Considering the ontology of DBpedia, the concept person has about 250 possible properties (properties from the superclass are excluded)¹¹. Therefore, the importance of a user assistance should be emphasized.

¹¹DBpedia person - <http://dbpedia.org/ontology/Person>

		Analec	BioNotate	Brat	GATE	Glozz	Inforex	Knowtator	MMAx2	Pundit	Vogon	WebAnno	Xconc
Feature-specific attributes													
A.1	Relationship annotation includes entity mention	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
A.2	Relationship annotation includes explicit relation mention	✓	✓	×	×	✓	×	×	×	×	✓	×	×
A.3	Relationship 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	\mathcal{C}	\mathcal{P}	\mathcal{C}	\mathcal{C}	\mathcal{P}	\mathcal{C}	\mathcal{C}	\mathcal{P}	\mathcal{C}	\mathcal{C}	\mathcal{C}	\mathcal{C}
A.8	Spatial positioning of the interface	\mathcal{F}	\mathcal{F}	\mathcal{F}	\mathcal{F}	\mathcal{R}	\mathcal{F}	\mathcal{F}	\mathcal{R}	\mathcal{F}	\mathcal{F}	\mathcal{F}	\mathcal{R}
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
<i>Fixed, Relative, Constant, Proportional</i>													

Table 6: Review results by attributes

References

- [1] Manuel Burghardt. *Engineering annotation usability - Toward usability patterns for linguistic annotation tools*. Pattern wiki: <http://www.annotation-usability.net>. 2014.
- [2] Manuel Burghardt. “Usability Recommendations for Annotation Tools”. In: *Proceedings of the Sixth Linguistic Annotation Workshop*. Jeju, Republic of Korea: Association for Computational Linguistics, 2012, pp. 104–112.
- [3] C. Cano et al. “Collaborative Text-annotation Resource for Disease-centered Relation Extraction from Biomedical Text”. In: *J. of Biomedical Informatics* 42.5 (Oct. 2009), pp. 967–977.
- [4] Paolo Ciccarese, Marco Ocana, and Tim Clark. “Open semantic annotation of scientific publications using DOME0”. In: *J. Biomedical Semantics* 3.S-1 (2012), S1.
- [5] R.P. Crawford. *The Techniques of Creative Thinking: How to Use Your Ideas to Achieve Success*. Hawthorn books. Hawthorn Books, 1954.
- [6] Aron Culotta, Andrew McCallum, and Jonathan Betz. “Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text”. In: *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. HLT-NAACL ’06. New York, New York: Association for Computational Linguistics, 2006, pp. 296–303.
- [7] Hamish Cunningham et al. *Developing Language Processing Components with GATE*. English. Version 8.0. The University of Sheffield, Department of Computer Science. 457 pp. June 22, 2016.
- [8] Julia Damerow, B.R. Erick Peirson, and Manfred D. Laubichler. “Don’t Panic! A research system for network-based digital history of science”. In: *Future of Historical Network Research Conference*. 2013.
- [9] Marco Grassi et al. “Pundit: Augmenting Web Contents with Semantics”. English. In: *Literary and Linguistic Computing* (2013. In Press).
- [10] Olaf Hartig. *Provenance Information in the Web of Data*. Ed. by Christian Bizer et al. 2009.
- [11] *HCI Patterns*. URL: <http://www.hcipatterns.org/patterns> (visited on 09/11/2016).
- [12] 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*. 2009, pp. 1–10.
- [13] Nancy Ide and Laurent Romary. “International Standard for a Linguistic Annotation Framework”. In: *Nat. Lang. Eng.* 10.3-4 (Sept. 2004), pp. 211–225.
- [14] Ali Khalili, Sören Auer, and Daniel 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.
- [15] J.-D. Kim et al. “GENIA corpora semantically annotated corpus for bio-textmining”. In: *Bioinformatics* 19.suppl 1 (2003), pp. i180–i182.
- [16] D.J. Kurland. *I Know What it Says... What Does It Mean?: Critical Skills for Critical Reading*. Brooks/Cole, 1994.
- [17] Frederic Landragin, Thierry Poibeau, and Bernard Victorri. “ANALEC: a New Tool for the Dynamic Annotation of Textual Data”. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey: European Language Resources Association (ELRA), 2012.
- [18] LDC. “ACE (Automatic Content Extraction) English Annotation Guidelines for Relations (Version 6.2)”. In: (2008).
- [19] W. John Macmullen. *Annotation as process, thing, and knowledge: Multi-domain studies of structured data annotation*. Tech. rep. in ASIST Annual Meeting, (Charlotte, NC, 2005), ASIST, in review, 2005.

- [20] Michał Marcińczuk, Jan Kocoń, and Bartosz Broda. “Inforex – a web-based tool for text corpus management and semantic annotation”. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. Ed. by Nicoletta Calzolari (Conference Chair) et al. Istanbul, Turkey: European Language Resources Association (ELRA), 2012.
- [21] Yann Mathet. *Glozz User’s Manual*. English. Version 1.0. GEYC. 79 pp. May 14, 2011.
- [22] Christoph Müller and Michael Strube. “Multi-level annotation of linguistic data with MMAX2”. In: *... and language pedagogy: New resources, new ...* (2006).
- [23] José Luis Navarro-Galindo and José Samos. “The FLERSA tool: adding semantics to a web content management system”. In: *IJWIS* 8.1 (2012), pp. 73–126.
- [24] Mariana Neves and Ulf Leser. “A survey on annotation tools for the biomedical literature”. In: *Briefings in Bioinformatics* (2012).
- [25] Philip V. Ogren. “Knowtator: A Protegé Plug-in for Annotated Corpus Construction”. In: *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume: Demonstrations*. NAACL-Demonstrations ’06. New York, New York: Association for Computational Linguistics, 2006, pp. 273–275.
- [26] Delip Rao, Paul McNamee, and Mark Dredze. “Multi-source, Multilingual Information Extraction and Summarization”. In: ed. by Thierry Poibeau et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Entity Linking: Finding Extracted Entities in a Knowledge Base, pp. 93–115.
- [27] Javier De Las Rivas and Celia Fontanillo. “Protein-Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks”. In: *PLoS Computational Biology* 6.6 (2010).
- [28] Pontus Stenetorp et al. “BRAT: A Web-based Tool for NLP-assisted Text Annotation”. In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. EACL ’12. Avignon, France: Association for Computational Linguistics, 2012, pp. 102–107.
- [29] *UI Patterns*. URL: <http://ui-patterns.com/patterns> (visited on 09/11/2016).
- [30] Andreas Vlachos and Mark Craven. “Biomedical event extraction from abstracts and full papers using search-based structured prediction”. In: *BMC Bioinformatics* 13.S-11 (2012), S5.
- [31] Antoine Widlöcher and Yann Mathet. “The Glozz Platform: A Corpus Annotation and Mining Tool”. In: *Proceedings of the 2012 ACM Symposium on Document Engineering*. DocEng ’12. Paris, France: ACM, 2012, pp. 171–180.
- [32] Seid Muhie Yimam et al. “WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1–6.

3 Additional Screenshots

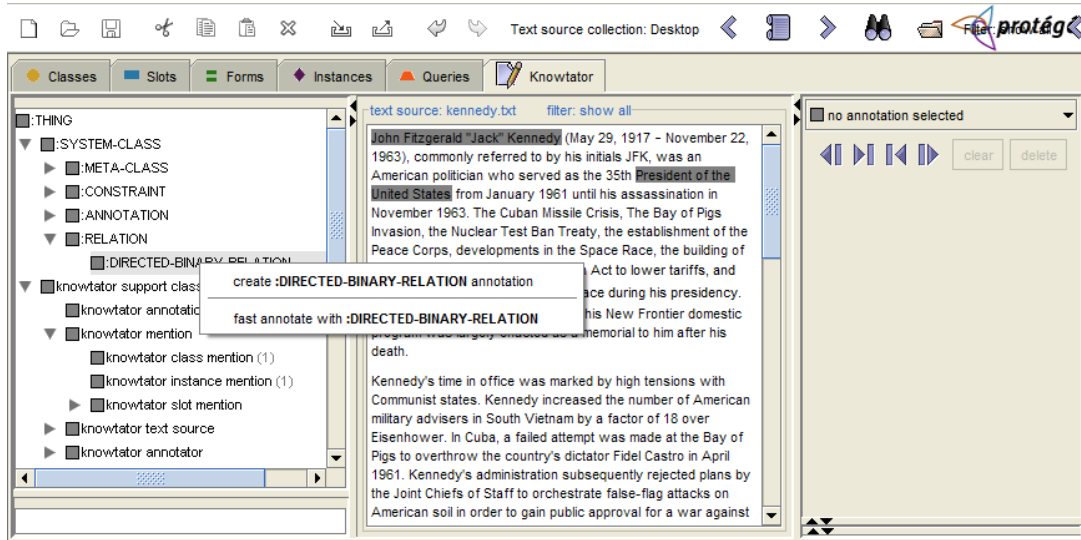


Figure 22: Create Relation in Knowtator

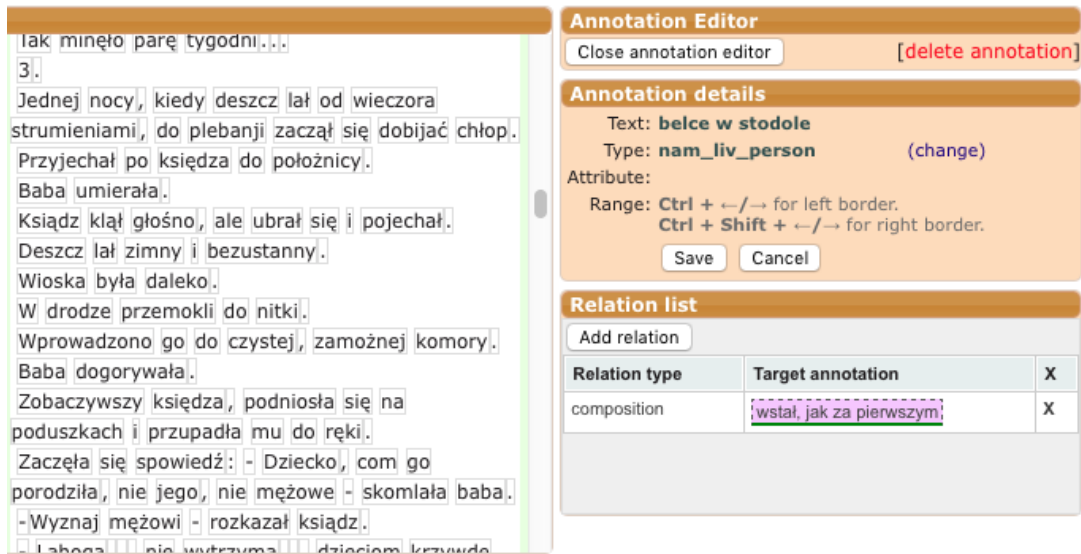


Figure 23: Relation List in Inforex

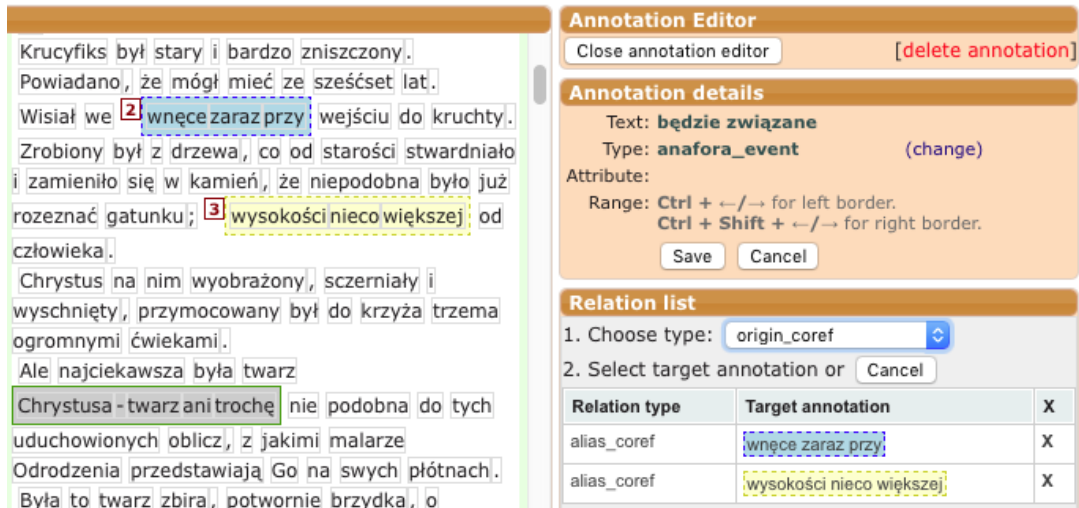


Figure 24: Visual Assistance in Inforex

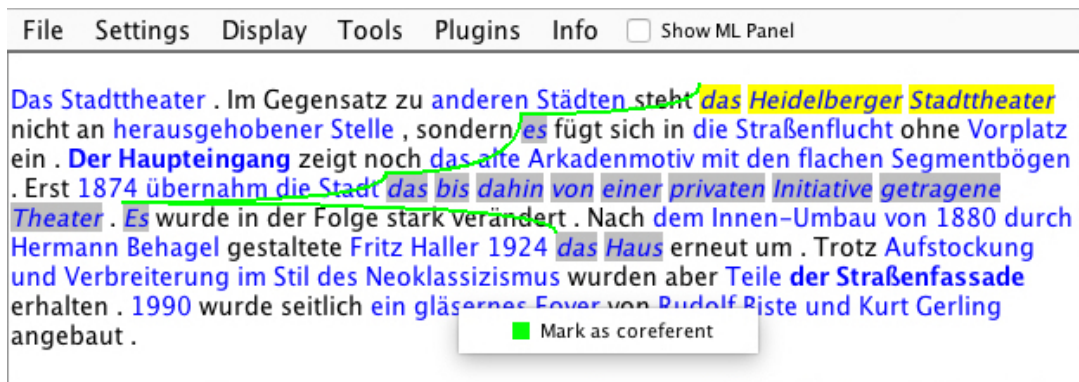


Figure 25: Relation Menu in MMAX2

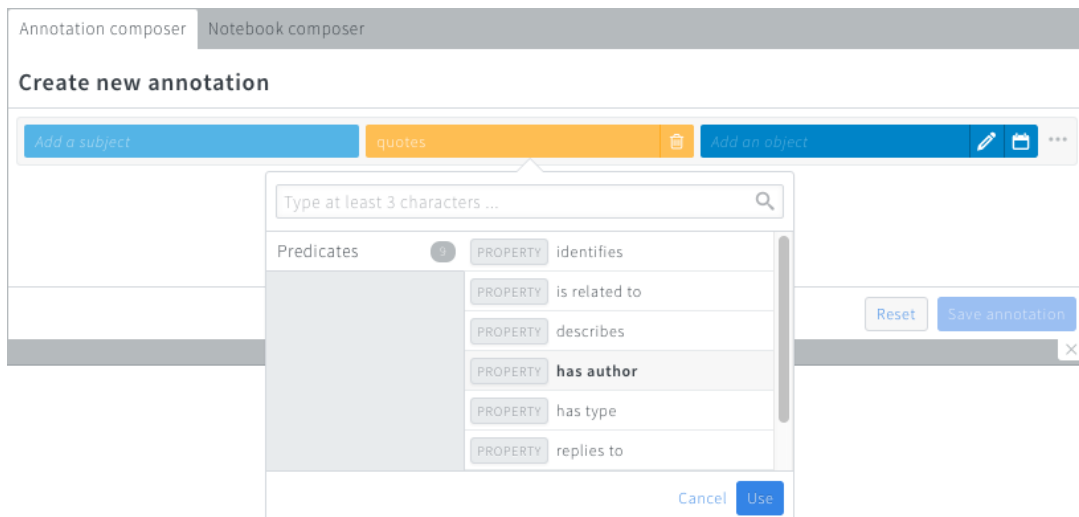


Figure 26: Predicate List in Pundit

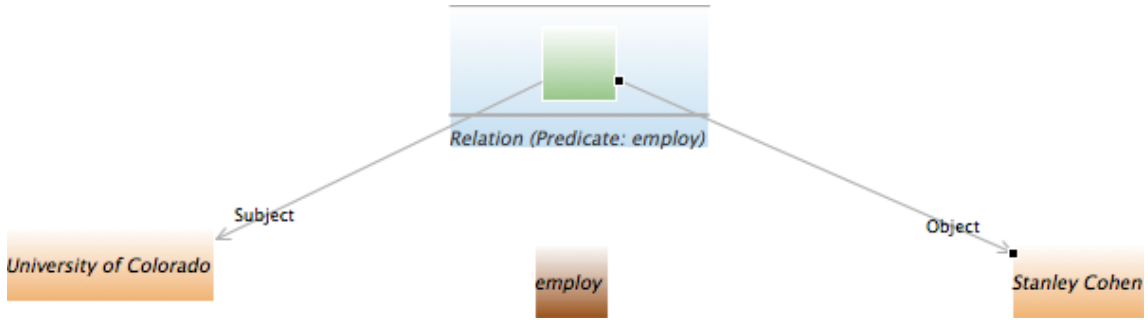


Figure 27: Visual Editor in Vohon

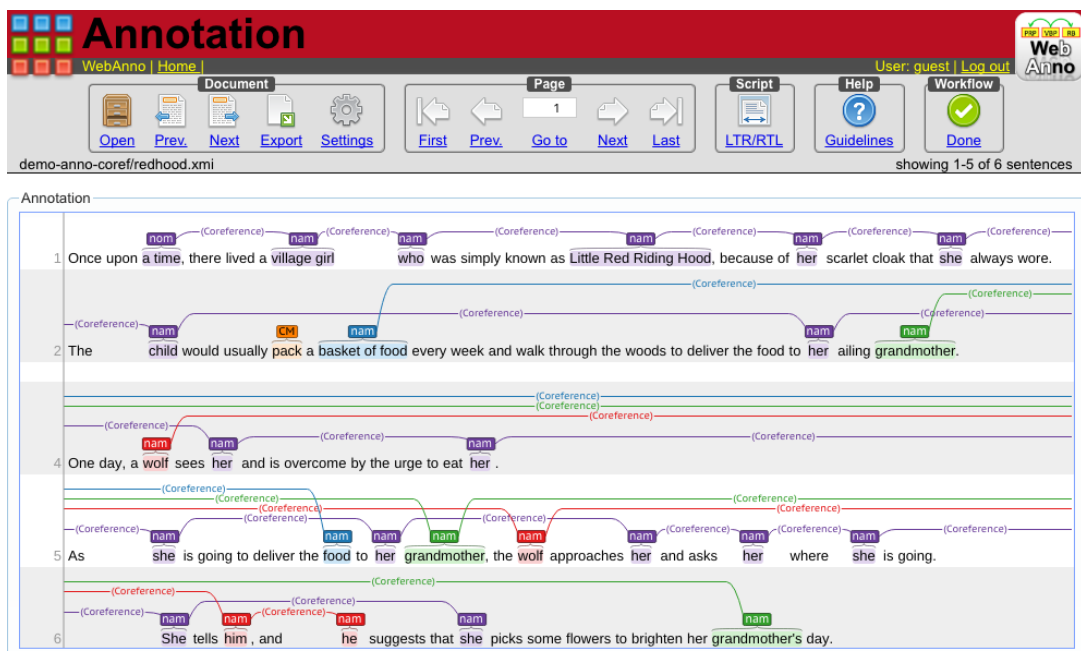


Figure 28: Interface of WebAnno

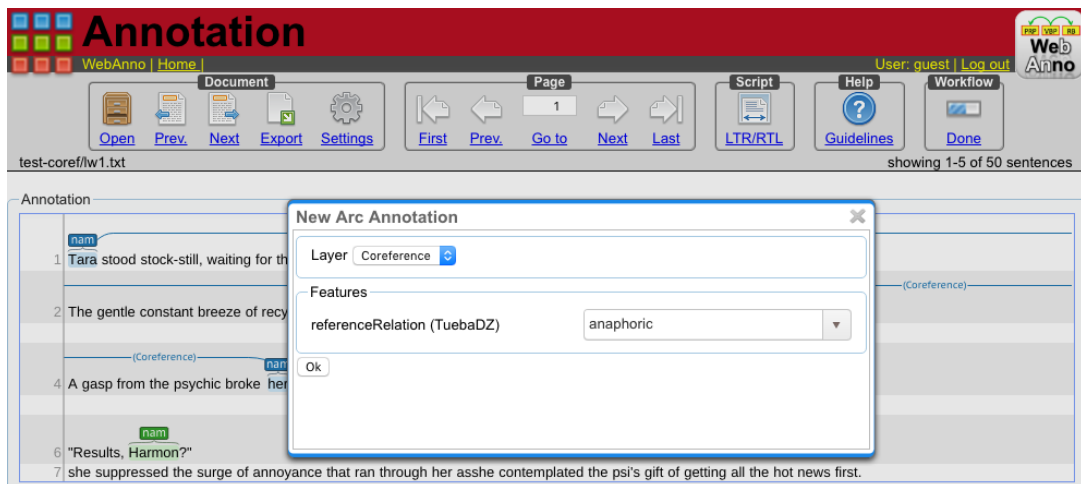


Figure 29: Relation Dialog in WebAnno

[2][a] the second-youngest president (after Theodore Roosevelt), and the first person born in the 20th century to serve as president.[3] To date, Kennedy has been the only Roman Catholic president and the only president to have won a Pulitzer Prize.[4]

Kennedy was assassinated in Dallas, Texas on November 22, 1963. Lee Harvey Oswald was arrested that afternoon and charged with the crime that night. Jack Ruby shot and killed Oswald two days later, before a trial could take place. The FBI and the Warren Commission officially concluded that Oswald was the lone assassin. The United States House Select Committee on Assassinations (HSCA) agreed with the conclusion that Oswald fired the shots which killed the president, but also concluded that Kennedy was probably assassinated as the result of a conspiracy.[5]

Figure 30: Screenshot of an existing relation

After military service as commander of Motor Torpedo Boats PT-109 and PT-59 during World War II in the South Pacific, Kennedy represented Massachusetts's 11th congressional district in the U.S. House of Representatives from 1947 to 1953 as a Democrat. Thereafter, he served in the U.S. Senate from that state from 1953 until 1960. Kennedy defeated Vice President and Republican candidate Richard Nixon in the 1960 U.S. Presidential Election. At age 43, he was the youngest man to have been elected to the office.

[2][a] the second-youngest president (after Theodore Roosevelt), and the first person born in the 20th century to serve as president. Kennedy has been the only Roman Catholic president and the only president to have won a Pulitzer Prize.[4]

Kennedy was assassinated in Dallas, Texas on November 22, 1963. Lee Harvey Oswald was arrested that afternoon and charged with the crime that night. Jack Ruby shot and killed Oswald two days later, before a trial could take place. The FBI and the Warren Commission officially concluded that Oswald was the lone assassin. The United States House Select Committee on Assassinations (HSCA) agreed with the conclusion that Oswald fired the shots which killed the president, but also concluded that Kennedy was probably assassinated as the result of a conspiracy.[5]

John F. Kennedy born on November 22, 1963

John F. Kennedy died on November 22, 1963

John F. Kennedy born on November 22, 1963

John F. Kennedy died on November 22, 1963

Lee Harvey Oswald was arrested that afternoon and charged with the crime that night. Jack Ruby shot and killed Oswald two days later, before a trial could take place. The FBI and the Warren Commission officially concluded that Oswald was the lone assassin. The United States House Select Committee on Assassinations (HSCA) agreed with the conclusion that Oswald fired the shots which killed the president, but also concluded that Kennedy was probably assassinated as the result of a conspiracy.[5]

Lee Harvey Oswald was arrested that afternoon and charged with the crime that night. Jack Ruby shot and killed Oswald two days later, before a trial could take place. The FBI and the Warren Commission officially concluded that Oswald was the lone assassin. The United States House Select Committee on Assassinations (HSCA) agreed with the conclusion that Oswald fired the shots which killed the president, but also concluded that Kennedy was probably assassinated as the result of a conspiracy.[5]

Creator: neonion-admin@fu-berlin.de

Creator: neonion-admin@fu-berlin.de

Figure 31: Screenshot of statement menu with existing relation