neonion – combining human and machine intelligence

Claudia Müller-Birn
Institute of Computer Science
Freie Universität Berlin, Germany
clmb@inf.fu-berlin.de

Alexa Schlegel
Institute of Computer Science
Freie Universität Berlin, Germany
alexa.schlegel@inf.fu-berlin.de

Tina Klüwer
Institute of Computer Science
Freie Universität Berlin, Germany
tina.kluwer@fu-berlin.de

André Breitenfeld
Institute of Computer Science
Freie Universität Berlin, Germany
andre.breitenfeld@fu-berlin.de

Abstract
The reading of text resources in scholarly settings can take various forms. Each form provides scholars with different insights that complement each other. The first findings from an ongoing series of interviews on scholarly annotation practices suggest that users are aware of the various forms of reading, but they are reluctant to use automatic annotations and still rely on conventional tools. In this paper, we introduce a prototype of annotation software that aims to interrelate different types of reading synergistically by employing a mixed-initiative approach.

Author Keywords
reading; annotation; digital humanities; mixed-initiative approach

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: Prototyping, Screen design.

Introduction
Reading is the most prevalent knowledge activity in scholarship (cp. [2]). Active or close reading, as it is commonly known, combines critical thinking with learning [9] and involves the annotation of documents [8] by highlighting, underlining text or adding comments [6]. Since schol-
ars increasingly read online (called “hyper reading” [10]), they also need to annotate these digital resources. Personal reading and the accompanying manual annotation is very time-consuming or, at worst, impossible given all the text resources that are now available. As a result, machine reading, "the automatic, unsupervised understanding” [3] is increasingly used to pre-process texts. Each of these forms of reading (close, hyper, or machine reading) provides scholars with different insights that complement each other [4]. Selected findings from an ongoing interview study suggest that scholars need all types of reading in their annotation practice, but a technically sound solution is still missing. In the following concept proposal, semantic annotation software is introduced that aims to synergistically interrelate the different types of reading (cp. [4]) by employing a mixed-initiative approach.

Selected findings from study on scholarly annotation practice

In order to understand to what extent scholars interrelate the different reading types, we are in the process of carrying out series of semi-structured interviews. The collected insights benefit an annotation software that is currently in development. In the following, we present selected findings from the first six interviews, conducted in person, of humanistic scholars (P1 to P6) from the area of history of science. We solicited participants by email, interviews lasted between one and one and a half hours and were transcribed verbatim for subsequent analysis. All participants are members of the same research institute. The institute plans to use the fully developed tool in a substantial and long term research project, in which scholars aim to identify concepts (e.g., persons, institutions, positions) in an extensive number of documents in order to support further analysis. Commonly, all interviewed scholars use tools such as Word or PDF software for annotating text, but admit existing disadvantages. As an example, one researcher (P1) states that her annotations are "buried" in PDF documents. Interviewees repeatedly mentioned the costs of manual annotation work, especially for recurring tasks such as named entity recognition or tagging (P1, P2, P3, P5). They regard automatic annotation as a starting point for further analyses: “[..] a basic stock of data can be created by automatic tagging.” (P2). Moreover, two participants suggest that automatic annotations should either be provided as suggestions or be generated as “basic” annotations that can be further adapted or extended by themselves (P3, P5). Despite these needs, scholars hesitate in using more appropriate software. Technological inexperience and an assumed steep learning curve: “this is great software […], but it is incredibly hard to learn.” (P2) impede its uptake. All participants gave statements in a similar vein. Our preliminary results suggest that the annotation software requires a simple to use automatic annotation support that allows for reusing and sharing annotation with other scholars. These requirements have already been built into a first software prototype called neonion1.

neonion – semantic annotation software

With neonion, we aim to provide a lightweight annotation tool for creating, sharing and reusing annotation data. The current prototype allows for the addition of semantic annotations to documents. Users are able to select text elements and describe those with a fixed vocabulary. One goal of the prototype is to hide the complexity of creating semantic annotations by providing an intuitive user interface that implicitly supports users in annotating texts. By concentrating on semantic annotations, we

1The name neonion is coined from the word “neon” for the color of a text marker and the word “onion” for emphasizing that annotations add additional layers of knowledge to a document.
can establish a test bed for synergistic interrelation between machine and hyper reading. In neonion, a mixed-initiative annotation concept is implemented that “explicitly support an efficient, natural interleaving of contribution by users and automated services” \[5\].

**Manual annotation concept**

In neonion, users can annotate phrases by referring to ontology concepts. As an example, scholars might want to annotate all people mentioned in historical documents, in order to then identify repeated occurrences of them in particular settings (e.g., a time and place). They could use the FOAF (Friend of a friend)-ontology in the annotator view, Person appears as one possible selector. We refer to this as concept level annotation (cp. Figure 1). Users are not only able to refer to a concept in an ontology, but can also link this concept to a directly identifiable resource on the Web. In neonion various URI’s can be used that represents this person as a resource on the Web. We refer to this as reference level annotation (cp. Figure 2).

**Semi-automated annotation concept**

neonion actively assists users in carrying out annotation tasks on both annotation levels. On the concept level, the software recommends possible annotations based on named entity recognition (NER)\(^2\). This recommendation process is visualized in Figure 3. Users can now work on the recommendations delivered by the software by accepting, rejecting or modifying the recommended annotations. The NER learns from user’s annotation decisions and improves subsequent recommendations accordingly.

\(^2\)A named entity recognizer automatically detects and classifies instances of entities in a text such as names of persons, organizations or locations.

Figure 1: Step 1–Concept level annotation with a pop-up that shows two user defined concepts (person and institution) and with the person concept selected.

Figure 2: Step 2 (optional)-Reference level annotation with a pop-up that appears after selecting a person concept from possible representations of people in Wikidata. The user can select the respective item, edit the search term, or mark the string as unknown for later annotation.

Figure 3: Sequence diagram of the annotation process showing three interaction possibilities.

As a result, additional entity models that improve the recognition performance for a particular document type are informed and trained by user feedback. On the reference level, the software automatically calculates appropriate entities that match the annotation on the basis of a selected knowledge base. In its current permutation, neonion offers an interface to Wikidata, allowing the software to automatically calculate the most suitable items in Wikidata that match the given concept type.

**Strategies for introducing algorithmic support**

Utilizing the settings above, users require annotating people in documents and have no previous experience in using an automated annotation service. As a conceptual starting point, three contexts for concept level annotations are specified that describe when the algorithm can take initiative and react to the user. For each context, it is defined how the software should notify the user about the possible algorithmic support offered (cp. Figure 4).
Of four different strategies (cp. [7]) to coordinate suggestions such as these, we selected two that were relevant to our scenario: those of negotiated and immediate interruptions. In the first context, users implicitly offer initiative to the software when their annotation task is a time-consuming routine. The usage of this context is anticipated if a user has been annotating the same concept for a period of time that is above the average usage time of the system, or the time between subsequent annotations (using the same concept) is well below average (for example, below the lower quartile of the distribution). A pop-up window immediately informs users about software support. In the second context, users take priority and give up initiative to the software when their annotation task becomes too time-consuming. This context is likely if users are steadily annotating concepts, but the time between subsequent annotations is far above average in similar annotation activities. This context occurs if users have difficulty in identifying concepts in the text and simultaneously need to search for additional information. Users are informed about available software support in two ways: (a) via a notifier in the margin of the document and (b) via icons near possible concepts. In the third context, if user annotating behavior varies, users are considered ambivalent (i.e. they either give or take the initiative). A concrete context is not identifiable, which recommends a less obtrusive software response – the negotiated coordination strategy.

**Figure 4:** Considered contexts for concept level annotations with the corresponding strategies for coordinating interruptions in each context (adapted from [1]).

**Conclusion and future work**

Initial findings from an interview study motivated the development of the prototype software described above, one that allows for interrelating hyper and machine reading in scholarly annotation. Interrelation such as this helps scholars to combine their insights more synergistically in the research process. In this contribution, a mixed-initiative approach is introduced that makes such synergistic interrelation possible. The theoretically derived mixed-initiative annotation concept needs to be evaluated next, in order to identify the most suitable strategy in suitably how to alert users.

**References**